

UNIVERSITÉ DU QUÉBEC À MONTRÉAL

UN MODÈLE HYBRIDE POUR LE SUPPORT À
L'APPRENTISSAGE DANS LES DOMAINES
PROCÉDURAUX ET MAL DÉFINIS

THÈSE

PRÉSENTÉE

COMME EXIGENCE PARTIELLE

DU DOCTORAT EN INFORMATIQUE COGNITIVE

PAR

PHILIPPE FOURNIER-VIGER

NOVEMBRE 2010

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

Avertissement

La diffusion de cette thèse se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.01-2006). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

REMERCIEMENTS

Je remercie tout d'abord Roger Nkambou, mon directeur de recherche, pour sa confiance, ses encouragements, ses précieux conseils et pour les nombreuses opportunités qu'il m'a offertes. Son support est l'élément clé qui a permis la réalisation de cette recherche.

Je suis également très reconnaissant envers André Mayers, mon codirecteur de recherche pour tout le temps qu'il m'a consacré depuis sept années, en me supervisant durant ma maîtrise et mon doctorat.

Je remercie aussi Engelbert Mephu Nguifo, professeur à l'Université Blaise-Pascal Clermont 2, avec qui j'ai eu le grand plaisir de travailler sur les aspects fouille de données de ma thèse, et qui m'a accueilli pendant un mois dans son laboratoire de recherche, en novembre 2009.

Je remercie également l'ensemble des membres du groupe de recherche GDAC et plus particulièrement Usef Faghihi avec qui j'ai collaboré sur plusieurs projets. Je remercie également Mikael Watrelot, Séverin Vigot et Lionel Tchamfong qui ont fait des stages au GDAC et m'ont aidé dans l'implémentation de mon travail pour son expérimentation.

J'adresse aussi des remerciements aux membres du jury, pour la lecture de cette thèse et leurs commentaires, ainsi que l'ensemble du personnel de l'Université du Québec à Montréal et plus particulièrement du Département d'informatique pour leur travail quotidien essentiel.

Je reconnais aussi la chance que j'ai eu d'obtenir une bourse de doctorat du Fonds Québécois de Recherche sur la Nature et les Technologies (FQRNT) ainsi qu'une bourse

Bourse d'études supérieures du Canada (BESC) du Conseil Recherche Sciences Nature et Génie (CRSNG). Sans ce financement et celui de mon directeur, la réussite de ce doctorat aurait été très difficile.

Je souhaite finalement présenter des remerciements à mes parents Jean Viger et Marie Fournier ainsi qu'à ma femme Zhuo Liu pour leur soutien constant tout au long de ce projet.

TABLE DES MATIÈRES

LISTE DES FIGURES	xi
LISTE DES TABLEAUX	xv
RÉSUMÉ	xix
INTRODUCTION	1
CHAPITRE I	
LES SYSTÈMES TUTORIELS INTELLIGENTS ET LES DOMAINES MAL DÉFINIS	9
1.1 Les systèmes tutoriels intelligents	9
1.1.1 Le module expert	11
1.1.2 Le modèle de l'apprenant	11
1.1.3 Le module tuteur	12
1.1.4 L'interface utilisateur	12
1.1.5 La représentation des connaissances dans les STI	13
1.2 Les domaines mal définis	13
1.2.1 Définition en intelligence artificielle	13
1.2.2 Définition de Lynch et al. dans le contexte des STI	14
1.2.3 Définition de Mitrovic et Weerasinghe dans le contexte des STI	16
1.3 Les approches pour offrir des services tutoriels dans les domaines mal définis	17
1.3.1 Le traçage de modèle	18
1.3.2 La modélisation à base de contraintes	19
1.3.3 L'incorporation d'un système expert	20
1.4 Les modèles pédagogiques pour offrir des services tutoriels dans les domaines mal définis	22
1.4.1 Structurer l'apprentissage autour de l'étude de cas	23
1.4.2 Supporter la métacognition	24
1.4.3 Supporter l'apprentissage par investigation	24
1.4.4 Utiliser des histoires interactives	26

1.4.5	Structurer l'apprentissage autour de la collaboration	27
1.4.6	Autres modèles pédagogiques	27
1.5	Le domaine d'application de la thèse	27
1.5.1	La manipulation robotique	28
1.5.2	CanadarmTutor, un STI pour la manipulation de Canadarm2	29
1.5.3	Comment donner une expertise du domaine à CanadarmTutor ?	30
1.5.4	L'importance du raisonnement spatial pour la manipulation de Canadarm2	32
1.6	Conclusion	32
CHAPITRE II		
UN MODÈLE COGNITIF POUR ÉVALUER LE RAISONNEMENT SPATIAL		35
2.1	Quelle est la nature des représentations spatiales ?	36
2.2	Comment implémenter les représentations spatiales dans un modèle informatique, et dans un STI ?	38
2.3	Le modèle cognitif de Fournier-Viger et al.	40
2.3.1	Le modèle théorique	41
2.3.2	L'implémentation du modèle	42
2.4	L'extension du modèle cognitif	43
2.5	Deux outils pour exploiter les aspects dynamiques du modèle	46
2.6	L'application du modèle étendu dans CanadarmTutor	50
2.7	Les services tutoriaux offerts avec le modèle étendu	52
2.7.1	Exploration libre des connaissances	52
2.7.2	Évaluation des connaissances par des exercices procéduraux	53
2.7.3	Évaluation des connaissances générales avec des questions	55
2.7.4	Génération d'indices, démonstrations et conseils personnalisés	58
2.7.5	Génération d'exercices personnalisés	59
2.7.6	Rétroaction immédiate et aide proactive	60
2.8	Discussion	61
2.9	Conclusion	62
2.9.1	Travaux futurs et extensions	64

CHAPITRE III	
LES APPROCHES D'ACQUISITION AUTOMATIQUE DE CONNAISSANCES DU DOMAINE	65
3.1 Les travaux sur l'acquisition automatique des connaissances en intelligence artificielle et en modélisation cognitive	66
3.1.1 Le travail de vanLent et Laird	66
3.1.2 Le travail de Langley et Ohlsson	66
3.2 L'apprentissage de connaissances du domaine dans les STI	67
3.2.1 Le système Demonstr8 de Blessing	67
3.2.2 Le Behavior Recorder de McLaren et al.	68
3.2.3 Le travail de Jarvis et Nuzzo-Jones	68
3.2.4 SimStudent de Matsuda et al.	69
3.2.5 Le travail de Barnes et Stamper	70
3.2.6 CAS de Suraweera et al.	70
3.2.7 Le travail de Riccuci et al.	71
3.3 Résumé comparatif des approches présentées et objectifs	71
CHAPITRE IV	
UNE APPROCHE POUR L'ACQUISITION AUTOMATIQUE DES CONNAIS- SANCES DU DOMAINE	75
4.1 La recherche de motifs séquentiels dans des traces d'utilisation	77
4.1.1 Le problème de la recherche de motifs séquentiels fréquents	77
4.1.2 La recherche de motifs séquentiels dans des traces d'utilisation	78
4.2 Extension de la recherche de motifs séquentiels avec des contraintes temporelles	80
4.2.1 La recherche de motifs séquentiels avec des contraintes temporelles	81
4.2.2 L'algorithme de Hirate et Yamana	82
4.2.3 La performance de l'algorithme de Hirate et Yamana	85
4.3 Extension de la recherche de motifs séquentiels avec le regroupement auto- matique de valeurs entières	86
4.3.1 L'extension	87
4.3.2 La performance de l'extension	89
4.4 Extension de l'algorithme de Hirate-Yamana pour extraire la représentation compacte des séquences fermées	90

4.4.1	Le BIDE	92
4.4.2	Le BackScan pruning	94
4.4.3	Intégration du BIDE et du BackScan Pruning à l'algorithme d'Hirate et Yamana	95
4.4.4	Compatibilité de BIDE+ avec le regroupement automatique des items avec valeurs	96
4.4.5	Performance de BIDE+	97
4.5	Extension de la recherche de motifs séquentiels avec une information sur le contexte	97
4.5.1	Algorithmes de recherche de motifs multidimensionnels	100
4.5.2	Recherche de motifs multidimensionnels fermés	102
4.6	Division des problèmes en sous-problèmes	103
4.7	Utilisation d'un minsup non constant	104
4.8	Les services tutoriels offerts sur la base des connaissances extraites	105
4.8.1	Évaluation du profil de l'apprenant	107
4.8.2	Suggestion d'indices personnalisés	108
4.8.3	Exploration libre des motifs	110
4.9	Évaluation de la performance dans CanadarmTutor	111
4.9.1	Caractéristiques des traces enregistrées	113
4.9.2	Résultats de l'extraction de motifs d'états	114
4.9.3	Résultats de l'extraction de motifs d'actions	115
4.9.4	Discussion et analyse des résultats	115
4.10	Autres applications des algorithmes de motifs séquentiels dans le domaine de l'AIED	116
4.11	Généricité de la solution	118
4.11.1	Généricité de l'approche pour d'autres STI	118
4.11.2	Généricité de l'algorithme en fouille de données	119
4.12	Conclusion	121
CHAPITRE V		
UNE EXTENSION DE L'APPROCHE AVEC DES RÈGLES SÉQUENTIELLES		123
5.1	La problématique de la recherche de motifs temporels pour la prédiction . .	124

5.2	La recherche de règles séquentielles : état de l'art	125
5.3	Notions préliminaires	128
5.4	Algorithme CMRules pour la recherche de règles séquentielles communes à plusieurs séquences	131
5.4.1	Preuve de complétude	132
5.4.2	Implémentation efficace de l'étape 3 de l'algorithme	134
5.4.3	Fusion des étapes 2 et 3 de l'algorithme	135
5.4.4	Analyse de la complexité de l'algorithme	135
5.4.5	Extensions de l'algorithme	136
5.5	Algorithme CMDeogun pour la recherche de règles séquentielles communes à plusieurs séquences	138
5.5.1	Notions préliminaires	138
5.5.2	Description de l'algorithme CMDeogun	140
5.5.3	Extensions de l'algorithme	142
5.6	Évaluation comparative des performances des algorithmes	144
5.6.1	Les bases de séquences utilisées	145
5.6.2	Première expérience : Kosarak-1	147
5.6.3	Deuxième expérience : Kosarak-2	147
5.6.4	Troisième expérience : BMS-WebView1	148
5.6.5	Quatrième expérience : Toxin-Snake	149
5.6.6	Cinquième expérience : Kosarak-1, $minSeqConf = 0.8$	151
5.6.7	Analyse des résultats	152
5.7	Intégration à l'approche d'acquisition automatique	154
5.7.1	Intégration dans le processus d'extraction des connaissances	154
5.7.2	Évaluation de la performance	157
5.7.3	Amélioration des services tutoriels	159
5.8	Généricité de la solution	160
5.8.1	Généricité de l'approche d'acquisition automatique pour d'autres STI	160
5.9	Généricité des algorithmes CMRules et CMDeogun en fouille de données	161
5.10	Conclusion	162

CHAPITRE VI	
UN MODÈLE HYBRIDE POUR LES DOMAINES MAL DÉFINIS	167
6.1 Le modèle hybride	169
6.1.1 Évaluation du profil de l'apprenant	169
6.1.2 Génération d'exercices personnalisés	171
6.1.3 Génération d'indices	171
6.1.4 Génération de conseils	172
6.1.5 Exploration libre des motifs et connaissances et démonstrations . . .	172
6.1.6 Rétroaction immédiate et aide proactive	173
6.2 L'évaluation empirique : cadre méthodologique	174
6.2.1 Les hypothèses	174
6.2.2 La conditions étudiée	175
6.2.3 Les personnes impliquées	175
6.2.4 La procédure	176
6.2.5 Prise en compte des considérations éthiques	177
6.3 L'évaluation empirique : les résultats	178
6.3.1 La pertinence des services tutoriels	178
6.3.2 L'impact des services tutoriels sur l'apprentissage	180
6.3.3 Autres commentaires des participants	180
6.3.4 Conclusion de l'expérience	180
6.4 Généricité du modèle hybride pour d'autres domaine	181
6.5 Conclusion	181
CONCLUSION	183

LISTE DES FIGURES

1.1	L'architecture classique des STI [163]	10
1.2	Les dimensions de mal définition selon Mitrovic et Weerasinghe	16
1.3	Le bras robotisé Canadarm2	30
1.4	CanadarmTutor	30
1.5	Un astronaute opérant Canadarm2 (NASA)	31
2.1	Attributs du modèle cognitif	43
2.2	Une structure de buts/sous-buts pour un exercice	49
2.3	Six sous-espaces	51
2.4	Le module « Knowledge Explorer »	53
2.5	Le fenêtre du modèle de l'apprenant	56
2.6	Les types de questions du « Space Quiz »	58
2.7	La génération de conseils, démonstrations et indices	59
4.1	(a) une base de 6 séquences et (b) des motifs séquentiels extraits	80
4.2	Algorithme pour calculer $ISDB (t, i)$	83
4.3	Algorithme de Hirate et Yamana	83
4.4	Sous-routine <i>algoProjection</i> de l'algorithme de Hirate et Yamana	83
4.5	Une application de l'algorithme de Hirate et Yamana (adaptée de [87])	84

4.6	Exemple de recherche de motifs séquentiels avec regroupement de valeurs	87
4.7	Exemple de recherche de motifs séquentiels fermés	91
4.8	Exemple de base de séquences multidimensionnelles	98
4.9	Exemple de motifs séquentiels multidimensionnels	98
4.10	Division de l'espace 3D en cubes	103
4.11	$\text{minsup}(x) = e^{(-0.4x-0.2)} + 0.2$	105
4.12	Algorithme <i>RecognizePlan</i>	106
4.13	Suggestion d'incide par l'approche d'acquisition automatique	110
4.14	Exploration libre des motifs	112
4.15	Un exercice de déplacement du bras robotisé	113
4.16	Nombre de nouveaux états pour chaque trace	114
5.1	Un exemple d'extraction de règles d'association	129
5.2	Un exemple d'extraction de règles séquentielles	131
5.3	Algorithme CMRules pour l'extraction de règles séquentielles	131
5.4	Un exemple d'exécution de l'algorithme CMRules	132
5.5	Algorithme CMDeogun pour l'extraction de règles séquentielles	143
5.6	Algorithme <i>expansionGauche</i>	143
5.7	Algorithme <i>expansionDroite</i>	144
5.8	Résultats de l'expérience avec Kosarak-1	148
5.9	Résultats de l'expérience avec Kosarak-2	149

5.10 Résultats de l'expérience avec BMS-WebView1	150
5.11 Résultats de l'expérience avec Toxin-Snake	150
5.12 Influence de <i>minSeqConf</i> avec Kosarak-1	151
5.13 Processus d'acquisition des connaissances avec l'ajout des règles	157
5.14 Nouvelle fenêtre pour la suggestion d'indice	160
6.1 Génération d'indices par le modèle hybride	173

LISTE DES TABLEAUX

2.1	Définition partielle du concept « CP7_vue_de_MPLM02 »	47
2.2	Définition partielle du concept « CP7_vue_de_MPLM01 »	47
2.3	Définition partielle de la procédure « RappelCaméraPourVueModuleSSI »	47
3.1	Tableau récapitulatif des approches présentées	73
5.1	Caractéristiques des jeux de données	145
5.2	Observations pendant les expériences	154

LISTE DES ACRONYMES

ACT-R : Adaptive Control of Thought - Rational

BIDE : Bi-Directional Extension closure checking

CTAT : Cognitive Tutor Authoring Tool

EAO : Enseignement Assisté par Ordinateur

LMS : Learning Management System

MIACE : Modèle Informatique de l'Acquisition des Connaissances par un Élève

MBC : Modélisation à base de contraintes

SE : Sous-espace

SSI : Station Spatiale Internationale

STI : Système Tutoriel Intelligent

RÉSUMÉ

Pour construire des systèmes tutoriels intelligents capables d'offrir une assistance hautement personnalisée, une solution populaire est de représenter les processus cognitifs pertinents des apprenants à l'aide d'un modèle cognitif. Toutefois, ces systèmes tuteurs dits cognitifs ne sont applicables que pour des domaines simples et bien définis, et ne couvrent pas les aspects liés à la cognition spatiale. De plus, l'acquisition des connaissances pour ces systèmes est une tâche ardue et coûteuse en temps. Pour répondre à cette problématique, cette thèse propose un modèle hybride qui combine la modélisation cognitive avec une approche novatrice basée sur la fouille de données pour extraire automatiquement des connaissances du domaine à partir de traces de résolution de problème enregistrées lors de l'usage du système. L'approche par la fouille de données n'offre pas la finesse de la modélisation cognitive, mais elle permet d'extraire des espaces problèmes partiels pour des domaines mal définis où la modélisation cognitive n'est pas applicable. Un modèle hybride permet de profiter des avantages de la modélisation cognitive et de ceux de l'approche fouille de données. Des algorithmes sont présentés pour exploiter les connaissances et le modèle a été appliqué dans un domaine mal défini : l'apprentissage de la manipulation du bras robotisé Canadarm2.

Mots-clés : Systèmes tutoriels intelligents, cognition spatiale, robotique, fouille de données

INTRODUCTION

La représentation des connaissances est un aspect fondamental de la conception de systèmes à base de connaissances. En effet, la création de tels systèmes nécessite une importante réflexion sur (1) les structures d'encodage des connaissances, (2) les processus d'acquisition de ces connaissances et (3) les mécanismes de raisonnement ou d'inférence qui les exploiteront. Ces trois aspects constituent des défis importants et reliés.

Dans le cadre des systèmes tutoriels intelligents (STI), un cas particulier de systèmes à base de connaissances qui vise à offrir une assistance personnalisée lors d'activités pédagogiques, cette problématique est encore plus importante. En effet, on retrouve dans ces systèmes plusieurs modules où l'explicitation de la connaissance est requis. Il est entre autres nécessaire de représenter les connaissances du domaine, pédagogiques ou liées à l'apprenant. Représenter les connaissances d'un domaine peut être facile pour certains domaines, mais très difficile pour d'autres, dépendant de la nature du domaine de connaissances, et du choix du formalisme et de l'approche d'acquisition des connaissances.

Les domaines où les approches classiques de représentation des connaissances pour construire des STI s'appliquent mal sont dits *mal définis*. Ils suscitent de plus en plus l'intérêt des chercheurs de la communauté comme en a témoigné l'organisation d'ateliers spécifiques sur ce sujet en 2006, 2007, 2008 et 2010 [6, 7, 8, 9]. Parce que de nombreux domaines de connaissances dans lesquels peuvent opérer les STI sont considérés mal définis, il s'agit d'une avenue de recherche importante. L'objectif principal poursuivi par ces recherches est de développer des approches alternatives aux approches classiques afin de faciliter le déploiement de STI dans ces domaines. Les travaux sur les domaines mal définis ont jusqu'à présent porté sur des domaines comme l'argumentation, les compétences interculturelles et le diagnostic médical [6, 7, 8]. Plusieurs solutions ont été proposées

pour offrir des services tutoriels dans ces domaines, certaines étant particulières à des domaines, d'autres tentant d'être plus générales, mais aucune n'est satisfaisante pour tous les domaines, puisqu'ils ne partagent pas tous les mêmes caractéristiques [105].

Problématique

De nombreuses problématiques liées à la conception de STI pour les domaines mal définis ont été récemment mises en évidence [105]. Cette thèse ne vise pas à s'attaquer à tous les défis : elle traite particulièrement des approches pour supporter les services tutoriels dans les domaines mal définis sur le plan de la représentation et de l'acquisition des connaissances du domaine nécessaires au fonctionnement d'un STI. Elle ne traite pas d'autres aspects comme par exemple le choix de stratégies pédagogiques appropriées pour enseigner les domaines mal définis. De plus, la thèse se restreint aux domaines axés sur la résolution de problèmes.

Cette thèse s'articule autour de deux éléments de problématique. Le premier part du constat que les approches classiques pour supporter les services tutoriels dans les STI (la modélisation à base de contraintes, le traçage de modèle et l'intégration de systèmes experts ; cf. chapitre 1) ne sont pas appropriées pour représenter les domaines mal-définis. La principale raison est qu'elles sont généralement conçues pour être appliquées à la main et qu'obtenir des représentations satisfaisantes des domaines mal définis de cette façon est impensable. Par conséquent, une question de recherche importante est : est-ce qu'il est possible de créer des approches pour automatiser l'acquisition des connaissances du domaine pour les domaines mal définis ? Certaines approches d'automatisation ont été proposées, par exemple pour extraire des modèles de tâches à partir de traces d'utilisation en appliquant divers algorithmes, mais toutes ont été conçues pour être appliquées à des domaines bien définis. De plus, toutes souffrent de limitations importantes. Par exemple, certaines requièrent de définir un ensemble minimal de connaissances du domaine, ne tiennent pas compte du profil des personnes qui enregistrent les traces d'utilisation, demandent d'annoter les traces d'utilisation ou doivent être appliquées pour chaque exercice (voir le chapitre 3 pour une revue de littérature détaillée).

Le second élément de problématique de cette thèse porte sur la dimension spatiale des tâches. Cet élément est motivé par le domaine d'application de la thèse qui est le domaine mal défini de la manipulation robotique, mais n'y est pas limitée. La manipulation robotique couvre de nombreux domaines comme l'opération de robot pour l'assemblage de voitures ou pour la détection de mines antipersonnelles [121]. La cognition spatiale joue un rôle crucial dans les tâches de manipulation ; par exemple lorsque des mouvements précis doivent être effectués avec un robot ou que les déplacements d'un robot doivent être planifiés, ou que des obstacles doivent être évités. De plus, le succès d'une tâche et sa bonne exécution sont souvent déterminés par des critères spatiaux comme l'atteinte d'un emplacement et la manipulation ou la transformation d'objets de l'environnement. Il est aussi souvent souhaité qu'un opérateur de robot se familiarise avec l'opération d'un robot dans un environnement ayant des propriétés spatiales particulières. Dans tous ces cas, la dimension spatiale est très importante pour l'accomplissement des tâches. Le deuxième élément de problématique se résume par la question de recherche suivante : comment donner les moyens à un STI d'évaluer le raisonnement et les représentations spatiales des apprenants dans le but de fournir des services tutoriels capables de favoriser leur acquisition ? Jusqu'à présent aucune recherche n'a étudié cette question importante.

Par ailleurs, la problématique intégratrice de cette thèse qui combine les deux éléments de problématique précédents relève de deux constats importants. D'une part, les solutions de modélisation cognitive habituellement utilisées pour représenter les domaines bien définis ne permettent pas de couvrir les aspects mal définis et d'explicitier les éléments liés à la cognition spatiale. D'autre part, même si l'automatisation de l'extraction de connaissances du domaine se présente comme une approche intéressante pour l'acquisition de connaissances dans certains domaines difficiles à formaliser, les modèles obtenus sont généralement moins précis que ceux créés par la modélisation cognitive (par exemple [91]). De ces constats dérive la question de recherche suivante, pertinente pour plusieurs domaines mal définis dont celui de la manipulation robotique : peut-on développer un modèle hybride combinant à la fois (1) une approche de modélisation cognitive fine de la tâche qui tient compte de la cognition spatiale et (2) une approche d'acquisition auto-

matique pour représenter d'autres parties de la tâche qui sont plus difficiles à formaliser, c'est-à-dire mal définis ?

Les hypothèses de la recherche

Les hypothèses de cette recherche sont au nombre de trois. Elles correspondent respectivement aux deux éléments de problématiques précédents et à la problématique intégrée.

Premièrement, pour faire face à la difficulté de formaliser certains domaines procéduraux et mal définis, une première hypothèse est que des techniques d'intelligence artificielle pourraient être appliquées pour automatiser l'acquisition des connaissances du domaine et que les connaissances extraites pourraient supporter des services tutoriels essentiels comme la suggestion d'indices et l'évaluation du profil de l'apprenant.

Deuxièmement, pour tenir compte de la dimension spatiale des tâches, une seconde hypothèse est qu'un modèle cognitif basé sur la recherche en cognition spatiale serait approprié. La justification d'utiliser un modèle cognitif dans un STI est que les modèles cognitifs ont le bénéfice d'offrir des modélisations fines [13]. Néanmoins, aucun modèle cognitif ne tient compte de l'aspect spatial.

Troisièmement, l'hypothèse intégratrice de cette thèse est que la représentation des connaissances dans certains domaines procéduraux et mal définis bénéficierait d'une approche hybride alliant à la fois les atouts (1) d'une approche de modélisation cognitive étendue permettant de modéliser finement certaines parties d'une tâche en tenant compte d'éléments comme la cognition spatiale, et celles (2) d'une approche d'extraction automatique permettant de représenter les aspects mal-définis du domaine.

Objectifs de la recherche

L'objectif global de la recherche est de développer un modèle de représentation des connaissances pour les STI. Ce modèle est hybride et comprend deux volets : (1) une approche originale d'acquisition automatique de connaissances du domaine pour les do-

maines procéduraux et mal définis et (2) un modèle original pour la modélisation cognitive incluant des éléments pour l'évaluation des connaissances spatiales. Deux objectifs spécifiques sont poursuivis par cette thèse.

Le premier objectif spécifique vise à développer une approche qui, contrairement aux approches proposées d'acquisition automatique ou semi-automatique de connaissances du domaine (présentées au chapitre 3) : (1) ne requiert aucune spécification préalable des connaissances du domaine ; (2) puisse tenir compte du profil de chaque apprenant ou expert effectuant une démonstration ou utilisant le système ; (3) puisse enrichir sa base de connaissances, s'il y a lieu, après chaque résolution de problème par un apprenant ; (4) soit applicable pour des domaines procéduraux comportant une explosion combinatoire de possibilités ; (5) ne suppose aucune vision restrictive de l'intelligence (comme des règles de production) et (6) ne demande pas aux utilisateurs de fournir des informations supplémentaires comme leurs intentions lors des démonstrations (comme cela se fait dans un processus d'analyse cognitive de tâches).

Le deuxième objectif spécifique de la thèse vise à utiliser le modèle hybride comme fondement des services tutoriels essentiels d'un STI cognitif à savoir : (1) l'évaluation des processus de résolution de problème d'un apprenant et (2) la génération/présentation d'indices, d'exercices et de démonstrations personnalisées. De plus, sur le plan de la cognition spatiale, le modèle devra permettre les services tutoriels suivant : (1) l'évaluation des représentations spatiales et habiletés spatiales d'un apprenant pertinentes à une tâche procédurale et (2) la génération d'exercices adaptés aux connaissances et habiletés spatiales d'un apprenant. Par ailleurs, le modèle devra être basé sur une théorie unifiée de la cognition (une théorie qui n'explique pas seulement un phénomène spécifique mais tente d'offrir une explication globale de la cognition) afin de permettre une évaluation intégrée du raisonnement spatial avec l'évaluation des processus de résolution de problème.

Les deux volets devront être intégrés dans un modèle hybride, être implémenté au sein d'un même STI, et être validé avec des apprenants pour démontrer sa capacité à offrir

une assistance pertinente favorisant l'apprentissage. Le STI choisi est CanadarmTutor [93, 124], un logiciel d'entraînement à la manipulation du bras canadien Canadarm2, une tâche complexe et mal définie, où le raisonnement spatial occupe un rôle crucial.

Méthode de recherche utilisée

La méthode utilisée pour cette recherche comporte 5 phases.

Phase 1 : Elle consiste dans un premier temps à définir la problématique de la thèse, en analysant les besoins de représentation des connaissances dans les domaines mal définis en général et en particulier dans CanadarmTutor, le domaine d'application de la thèse. En second lieu, elle consiste à effectuer une revue de littérature détaillée des modèles cognitifs de la cognition spatiale et des approches d'acquisition automatique de connaissances du domaine.

Phase 2 : Dans cette phase est effectué le développement d'un modèle cognitif. La première étape est d'étudier les différents modèles cognitifs pour les STI et d'explorer les travaux de recherche sur la cognition spatiale. La deuxième étape consiste en la proposition d'un modèle cognitif et l'intégration de ce dernier dans CanadarmTutor.

Phase 3 : Cette phase consiste à développer une nouvelle approche d'acquisition automatique de connaissances du domaine. Les différentes approches d'acquisition automatique ou semi-automatique de connaissances du domaine pour la résolution de problème sont étudiées et leurs caractéristiques et limites sont identifiées. Un ensemble d'objectifs est établi pour le développement d'une nouvelle approche. La nouvelle approche est spécifiée, développée et intégrée à CanadarmTutor. Des services tutoriels sont développés pour montrer l'utilité des modèles de connaissances construits par l'approche.

Phase 4 : Dans cette phase, le modèle cognitif et l'approche d'acquisition automatique des connaissances sont combinés dans un modèle hybride pour tirer profit du meilleur des deux approches. Le modèle hybride est mis en œuvre dans le logiciel

CanadarmTutor.

Phase 5 : La cinquième phase est l'évaluation/validation des contributions de la thèse.

L'évaluation est faite selon les angles suivants :

E1 : Évaluation formelle des algorithmes

E1.1 Analyse de complexité

E1.2 Preuve de théorèmes

E2 : Mesures de performance des algorithmes

E2.1 évaluation interne des capacités, et en particulier l'aptitude à traiter de larges volumes de données.

E2.2 évaluation comparative avec des algorithmes de la littérature ayant une fin similaire

E3 : Évaluation de l'utilité de la solution

E3.1 Évaluation empirique de l'utilité des services tutoriels offerts avec des utilisateurs

E3.2 Évaluation de la généricité des solutions proposées au niveau du domaine des STI.

E3.3 Évaluation de la généricité des solutions proposées au niveau d'autres domaines (par exemple pour d'autres types de problèmes dans le domaine de la fouille de données).

E3.4 Comparaison sur un plan théorique des propriétés des solutions proposées avec celles des approches alternatives pour le domaine des STI.

Pour évaluer le modèle cognitif, les angles de validation E3.2 et E3.4 sont employés. Pour évaluer les algorithmes d'acquisition automatique les angles E1, E2, E3.2 et E3.3 sont utilisés. Pour évaluer les services tutoriels offerts grâce à l'approche d'acquisition automatique, les angles E3.2 et E3.4 sont utilisés, respectivement. Finalement, l'évaluation du modèle hybride est faite selon les angles E3.1, E3.2 et E3.4. Il est à noter

qu'une évaluation empirique de l'utilité des services tutoriels est réalisée seulement pour le modèle hybride (E3.1).

Organisation de la thèse

La thèse comprend 7 chapitres.

Le chapitre 1 identifie tout d'abord les principales composantes des STI, et présente la problématique posée par les domaines mal définis et les solutions utilisées jusqu'à présent pour construire des STI pour ces domaines. Finalement, ce chapitre présente le domaine d'application de la thèse qui est la manipulation robotique, ainsi que le STI CanadarmTutor.

Le chapitre 2 offre une revue de littérature des modèles cognitifs de la cognition spatiale et leurs limites pour leur application dans les STI. Puis il propose un modèle cognitif pour construire des STI tenant compte de la dimension spatiale des tâches.

Le chapitre 3 présente une revue des approches d'acquisition automatique ou semi-automatique de connaissances d'un domaine.

Le chapitre 4 propose une nouvelle approche d'acquisition automatique des connaissances basée sur un algorithme d'extraction de motifs séquentiels.

Le chapitre 5 propose une extension de la solution proposée au chapitre 4 avec un algorithme d'extraction de règles séquentielles.

Le chapitre 6 décrit le modèle hybride combinant le modèle cognitif et l'approche d'extraction automatique de connaissances du domaine. Puis il présente une évaluation empirique du modèle hybride dans CanadarmTutor.

Finalement, la conclusion résume les contributions de la thèse et présente des perspectives pour la poursuite de la recherche.

CHAPITRE I

LES SYSTÈMES TUTORIELS INTELLIGENTS ET LES DOMAINES MAL DÉFINIS

Ce chapitre présente tout d’abord le contexte général de cette thèse qui est celui des systèmes tutoriels intelligents (STI). Puis il décrit la problématique des domaines dit mal définis, les domaines où les approches classiques pour construire des STI s’appliquent mal ou ne s’appliquent pas. Ce chapitre offre une revue des solutions proposées pour construire des STI pour ces domaines en notant leurs avantages et limitations. Finalement, le chapitre présente CanadarmTutor, le système tutoriel intelligent utilisé comme cadre d’application de la thèse, et indique en quoi ce STI opère dans un domaine mal-défini.

1.1 Les systèmes tutoriels intelligents

La première génération de système d’enseignement par ordinateur est apparue dans les années 1970. Ce sont les systèmes d’enseignement assisté par ordinateur (EAO). Ils proposent des activités éducatives simples sur ordinateur telles que des questions à choix multiples. Les concepteurs prédéfinissent la totalité des rétroactions : les décisions pédagogiques sont statiques et l’enseignement n’est pas individualisé [42]. Par conséquent, tout apprenant qui commet une erreur dans une activité d’apprentissage recevra la même rétroaction préétablie du système. Les STI pallient théoriquement les limites mentionnées des logiciels EAO. L’idée fondamentale des STI est d’offrir un enseignement personnalisé, dans le but de favoriser l’apprentissage. Pour accomplir cet objectif, les connaissances

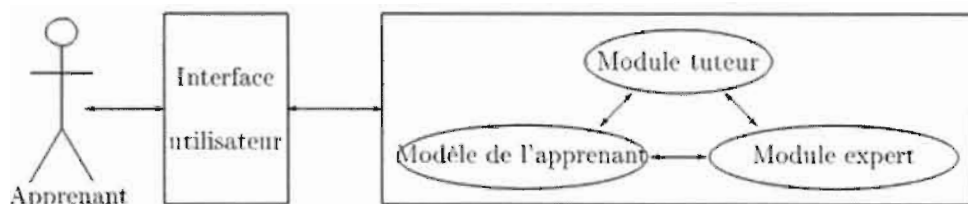


Figure 1.1 L'architecture classique des STI [163]

pédagogiques et les connaissances du domaine sont encodées séparément et un module dédié à l'évaluation des connaissances de l'apprenant est intégré. Un STI exploite ces trois types de connaissances par des stratégies pédagogiques pour bâtir un enseignement dynamique [163].

Les STI et EAO peuvent être vus comme des sous-classes des systèmes de formation en ligne. En fait, les STI et EAO ne proposent d'ordinaire que de courtes activités qui correspondent à la mise en pratique d'une théorie acquise en classe [159]. Cette portée restreinte est ce qui facilite la construction de STI, car les construire demande souvent de modéliser finement un domaine, ce qui est généralement très coûteux en temps. Par opposition, une autre classe importante de systèmes de formation en ligne nommés LMS (Learning Management Systems) sont généralement employés pour enseigner des cours complets et offrent une très faible personnalisation (ex. : cours d'université à distance). Ces derniers font généralement appel à l'intervention de formateurs humains [89], à l'antipode des STI qui ne proposent généralement, que des activités entre un apprenant et un logiciel [163, 159]. Notre étude porte sur les STI. Plusieurs architectures de STI ont été proposées. Les prochains paragraphes présentent brièvement l'architecture classique (cf. fig. 1.1), formée des composantes suivantes : un module expert, un modèle de l'apprenant, un module tuteur [163] et une interface utilisateur. La description de base des quatre modules qui suit, s'appuie sur le texte de Wenger [163] quoiqu'enrichie par quelques textes plus récents. La description est fonctionnelle, car de nombreuses variations et choix d'implémentation sont possibles.

1.1.1 Le module expert

Un STI doit, tout comme un enseignant humain, maîtriser les notions qu'il enseigne. C'est le rôle du module expert. Dans ce module, les connaissances du domaine sont encodées explicitement. Elles doivent être suffisamment précises pour que le système soit capable de générer la solution d'un expert humain. La simulation d'un expert humain sur un problème du domaine, permet (1) de montrer les étapes de résolution du problème ou sa solution à l'apprenant et (2) de comparer les actions de l'apprenant pour déterminer si sa solution est correcte ou s'il est dans la bonne voie [33].

1.1.2 Le modèle de l'apprenant

Le modèle de l'apprenant, indispensable à la personnalisation, contient la représentation par le système de l'apprenant. Ce modèle est construit et maintenu en recueillant des données sur l'apprenant et en appliquant des inférences sur ces données. Un modèle de l'apprenant ne couvre que des aspects pertinents à la réalisation des activités pédagogiques. La nature du modèle de l'apprenant dépend fortement des besoins du module tuteur. Un tel modèle pourrait contenir les informations suivantes : des degrés de certitude sur la maîtrise de chaque notion, les préférences de l'apprenant, les intentions ou émotions perçues de l'apprenant, un historique des actions de l'apprenant et un historique des résultats de l'apprenant.

Le modèle de l'apprenant emploie généralement les mêmes structures de représentation des connaissances que le module expert afin de faciliter la mise en correspondance des éléments de connaissance de ces deux modules. Trois approches prédominent pour mettre en relation les connaissances du modèle de l'apprenant avec celles du module expert [158]. Premièrement, le modèle par recouvrement représente le savoir de l'apprenant comme un sous-ensemble de celui de l'expert. Ce modèle attribue les erreurs de l'apprenant à l'ignorance d'éléments de connaissance nécessaires à l'accomplissement d'un but [27]. Deuxièmement, le modèle différentiel consiste à effectuer un recouvrement sur le sous-ensemble des connaissances correspondant aux objectifs de la formation plutôt que sur

l'ensemble des connaissances de l'expert [94]. Troisièmement, le modèle par perturbation exprime les connaissances de l'apprenant comme l'ensemble des connaissances de l'expert augmenté de connaissances erronées dont l'utilisation mène à l'erreur [25]. Dans les trois approches, l'encodage des connaissances doit être suffisamment précis pour représenter le manque de connaissances ou les conceptions erronées, sans toutefois modéliser des détails inutiles.

Le principal bénéfice de l'inclusion d'un modèle de l'apprenant réside dans la capacité à donner des explications et à générer des problèmes en fonction d'une hypothèse sur des connaissances acquises par l'apprenant. Basé sur un modèle de l'apprenant, le module tuteur peut intervenir aux moments propices tels que certaines approches éducationnelles le prescrivent et non pas seulement lorsque l'apprenant commet une erreur. En effet, juger les circonstances idéales d'intervention nécessite la connaissance de l'état des acquis de l'apprenant.

1.1.3 Le module tuteur

Le module tuteur prend les décisions pédagogiques. Ses trois tâches principales sont : (1) de planifier et d'adapter une séquence appropriée d'activités pédagogiques, (2) de choisir la nature des interactions et (3) de déterminer les circonstances idéales pour intervenir [163]. Ces décisions sont généralement fondées sur un encodage explicite des principes pédagogiques, didactiques et communicationnels (connaissances pédagogiques). Le module tuteur a recours au module expert et au modèle de l'apprenant pour mener à bien ces tâches.

1.1.4 L'interface utilisateur

L'interface utilisateur gère la communication avec l'apprenant. Elle offre une vue graphique sur le système et des outils pour le manipuler. Différents types d'interface existent : interfaces sous forme de pages Web accessibles avec un navigateur, interface de type « micro-monde », etc. (voir [164, p.39] pour une revue détaillée).

La conception adéquate d'une interface utilisateur constitue un facteur critique de l'efficacité d'un STI puisque l'apprenant ne peut pas recourir à la connaissance du domaine pour comprendre le fonctionnement de l'interface. Une interface intuitive [73], bien conçue et adaptée au niveau de développement de l'apprenant, lui permettra d'allouer davantage de ressources cognitives au problème à résoudre [23, 73].

1.1.5 La représentation des connaissances dans les STI

Comme pour tout système à base des connaissances, les connaissances des STI sont cruciales à leur fonctionnement. Elles nourrissent les différents modules : modèle de l'apprenant, modèle de l'expert et module tuteur. Ainsi, différents types de connaissances peuvent être prévus telles que les connaissances du domaine, pédagogiques, didactiques (connaissances pédagogiques pour enseigner des connaissances particulières d'un domaine), et des connaissances sur des apprenants construites dynamiquement. Pour chacun de ces types de connaissances, différentes structures d'encodage et d'algorithmes pour les exploiter permettent différents degrés d'interaction.

1.2 Les domaines mal définis

Dans le domaine des STI, de plus en plus de recherches portent sur la construction de STI pour des domaines de connaissances dits « mal définis » [6, 7, 8]. Ces domaines suscitent de l'intérêt, car ils sont nombreux et ils posent de nouveaux défis pour la construction de STI. Cette section aborde la question de ce qu'est un domaine mal défini. Puis, les deux sections suivantes présentent les différentes solutions proposées pour construire des STI pour ces domaines, et leurs avantages et limitations.

1.2.1 Définition en intelligence artificielle

La notion de domaines mal-définis prend ses racines en intelligence artificielle. Simon [141] en a proposé la définition sur la base de l'étude de la résolution de problèmes chez l'humain et de la construction de logiciels dotés de capacités de résolution de problèmes.

Pour Simon, un domaine mal défini est un domaine contenant des problèmes mal structurés. Un problème mal-structuré est un problème possédant au moins une des propriétés suivantes [141] :

Point de départ flou. Les instructions ou informations nécessaires pour débiter la résolution du problème sont incomplètes ou vagues.

But imprécis. Le critère qui détermine si le but est atteint est complexe et imprécis. Il peut y avoir plusieurs solutions discutables [105].

Absence de stratégies pour obtenir une solution. Il n'y a de stratégies pour déterminer les prochaines actions à effectuer à chaque étape du processus de résolution.

La définition de Simon peut être appliquée dans le contexte des STI puisque résoudre un exercice offert par un STI peut être vu comme résoudre un problème.

1.2.2 Définition de Lynch et al. dans le contexte des STI

Toutefois, l'engouement pour les domaines mal définis dans le contexte des STI a véritablement débuté en 2006, lors d'un atelier sur le sujet. Lynch et al. [105] ont alors offert une définition de travail de ce qu'est un domaine mal défini pour les STI. Cette définition englobe celle de Simon. Lynch et al. l'ont proposée sur la base d'une vaste revue de littérature des travaux du domaine des STI, de l'intelligence artificielle et de la prise de décision dans l'incertitude. Cela a amené Lynch et al. à conclure que les domaines mal définis possèdent une ou plusieurs des caractéristiques suivantes [105].

Solutions multiples et discutables. Les domaines où les problèmes ont plusieurs solutions discutables et aucune procédure claire pour évaluer les solutions sont mal définis. Un exemple de domaine est la conception de diagrammes entité relation à partir de descriptions textuelles. Il y a un nombre potentiellement infini de diagrammes respectant une description et le processus d'évaluation des diagrammes est subjectif. Un deuxième exemple est l'argumentation légale. Plusieurs arguments légaux peuvent être proposés pour un cas légal et il n'y a pas qu'une seule

bonne solution, bien que certaines peuvent être préférées selon des critères comme l'esthétique ou le succès [105].

Aucune théorie formelle du domaine. Les domaines sans théorie claire et complète pour déterminer les étapes de résolution des problèmes et tester la validité des solutions sont mal définis. Par exemple, pour les domaines de la composition musicale et de l'architecture, il y a seulement des théories incomplètes. Ceci contraste avec des domaines bien définis comme la géométrie.

Structure de tâche mal définie. De la perspective des tâches, trois grands types de domaines sont identifiés par Lynch et al. Les deux premiers sont mal définis. Premièrement, les « domaines de conception » offrent des tâches où l'objectif est de créer de nouveaux artefacts (par exemple, écrire une histoire ou composer de la musique). Ces domaines sont mal définis car le but est l'originalité, bien que certaines règles ou principes puissent guider la création. Le deuxième grand type de domaine est celui des domaines analytiques où il n'y a pas une façon certaine de bien faire l'analyse. L'absence de façon certaine survient fréquemment lorsque les données sont floues, incomplètes, fausses ou bien portent sur un environnement dynamique. Pour cette raison, ils sont dits mal définis. Des exemples sont l'achat de titres boursiers et le diagnostic médical. Le troisième type de domaine est celui des domaines basés sur la résolution de problèmes. Ces domaines requièrent d'appliquer une théorie formelle pour résoudre des problèmes ayant une réponse précise et vérifiable. De tels domaines sont considérés bien définis. Des exemples sont les problèmes de mathématiques qui ne demandent pas la construction de nouvelles connaissances (par exemple, calculer le volume d'une sphère à partir de son rayon).

Concepts abstraits ou indéterminés. Les concepts abstraits ou partiellement indéterminés sont problématiques quand ils doivent être appliqués dans des situations concrètes. Les domaines incluant de tels concepts sont mal définis. Ceci inclut la plupart des domaines utilisant le langage naturel parce que les mots et les phrases peuvent être ambigus. Un autre exemple est le droit, où plusieurs concepts sont abstraits et peuvent avoir plusieurs interprétations.

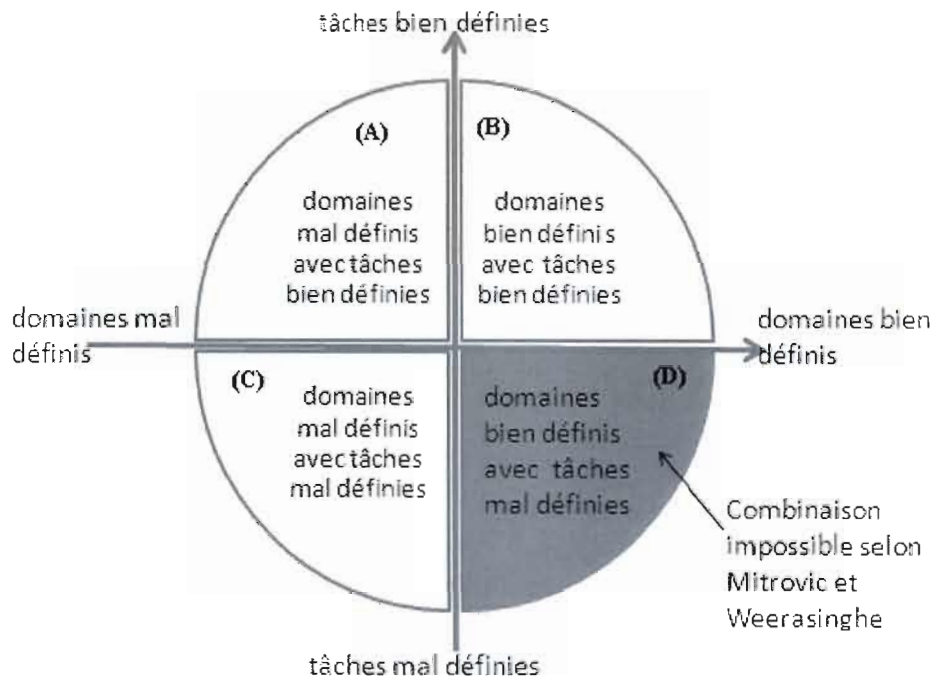


Figure 1.2 Les dimensions de mal définition selon Mitrovic et Weerasinghe

Sous-problèmes dépendants. Les domaines incluant des problèmes qui ne sont pas décomposables en sous-problèmes indépendants plus faciles à résoudre, sont aussi dits mal définis. Un exemple est celui de la construction d'une maison. Cette tâche demande de choisir un site approprié pour la construction et aussi un plan de construction, mais ces deux tâches sont dépendantes. Par conséquent, elles ne peuvent pas être effectuées séparément pour réduire la difficulté de la tâche sans compromettre en compromettre le succès.

1.2.3 Définition de Mitrovic et Weerasinghe dans le contexte des STI

Dans le domaine des STI, récemment, Mitrovic et Weerasinghe [115] ont considéré la mal définition des domaines sous deux dimensions : les tâches et les domaines. Ces deux dimensions peuvent être mal définies ou bien définies, de sorte qu'il y a quatre combinaisons de tâches/domaines. Ces combinaisons sont illustrées à la figure 1.2 par

les quatre quadrants. Dans le cadre de cette division, la dimension domaine est formée par les connaissances déclaratives ou la théorie du domaine qui peuvent être utilisées dans une tâche [115].

La proposition de Mitrovic et Weerasinghe comporte toutefois plusieurs problèmes. Premièrement, Mitrovic et Weerasinghe ont suggéré qu'il n'existe pas de domaine mal défini contenant des tâches bien définies [115], ou en d'autres mots que le quadrant (d) de la figure 1.2 est vide. Par conséquent, la dimension du domaine est discutable, spécialement du point de vue des chercheurs du domaine des STI, puisque le but de construire des STI est principalement d'assister les apprenants dans des tâches individuelles [164]. En effet, pour construire un STI, seul les tâches et les connaissances doivent être considérées dans la sélection des techniques appropriées pour offrir des services tutoriels. De plus, parce qu'un domaine peut contenir autant des tâches bien définies que mal définies, il ne semble pas pertinent de parler de la mal définition d'un domaine en entier. Par exemple, pour le domaine du génie logiciel, un STI pourrait offrir des questions à choix multiples sur les propriétés des diagrammes UML, mais aussi des problèmes mal définis comme celui de concevoir un diagramme UML.

Lynch et al. [105] avait d'ailleurs précisé qu'ils ont utilisé le mot « domaine » plutôt que « problème » simplement « pour mettre l'accent sur le fait que le but de l'enseignement est généralement des connaissances générales d'un domaine plutôt que des réponses spécifiques à un problème », et que « la distinction entre problèmes et domaines est immatérielle » [105]. Pour toutes ces raisons, la suite de cette thèse n'utilise que les définitions de Lynch et al. et Simon pour caractériser les domaines mal définis.

1.3 Les approches pour offrir des services tutoriels dans les domaines mal définis

Il y a trois approches classiques pour fournir des services tutoriels dans les STI axés sur la résolution de problèmes. Cette section les présente tout en illustrant leurs limites et avantages pour les domaines mal définis.

1.3.1 Le traçage de modèle

La première approche est le traçage de modèle. Les STI utilisant le traçage de modèle sont généralement conçus par analyse cognitive de tâche. Ce processus consiste à produire des espaces problèmes effectifs ou modèles de tâche en observant des utilisateurs novices ou experts [97] pour capturer les différentes façons de résoudre un problème.

Un modèle de tâche peut être créé pour un problème ou un ensemble de problèmes. Les modèles de tâche sont généralement représentés par un ensemble de règles de production (parfois structurées en un *arbre de décomposition de sous-but*s [164], p. 90) ou un espace d'états [10], où chaque règle ou transition correspond à une action ou une opération pour effectuer une tâche. Lorsque le formalisme utilisé est basé sur des théories cognitives, les STI produits sont appelés des « tuteurs cognitifs » [13]. La création d'un espace problème ou modèle de tâche permet d'offrir plusieurs services tutoriels. Tout d'abord, elle peut permettre de suivre le processus de résolution de problème d'un apprenant parmi plusieurs cheminements possibles (le *traçage de modèle*). Cela peut permettre à un STI de lui suggérer la prochaine étape à accomplir. Dans le cas où le modèle de tâche est annoté avec des ressources didactiques comme des indices, ces derniers peuvent être présentés à l'apprenant pour l'aider (par exemple dans Koedinger et al. [96]). On retrouvera aussi souvent la présence de connaissances ou d'habiletés correctes ou erronées (qui annoteront un espace problème, par exemple dans Koedinger et al. [96]), ce qui permettra le diagnostic des connaissances de l'apprenant par le suivi de ses actions. D'autres services tutoriels qui peuvent être offerts sur la base du traçage de modèle sont la génération de démonstrations et la reconnaissance des intentions d'un apprenant [164].

Les STI basés sur le traçage de modèle sont recommandés pour les tâches où il est souhaité d'évaluer le processus pour accomplir une tâche plutôt que le résultat. Cette approche est fondée sur une hypothèse forte qui considère que le point de départ (l'état initial), le but d'un problème (l'état but) ainsi que la démarche de solution sont bien définis. Cette hypothèse fausse l'applicabilité de cette approche dans les domaines mal

définis si on s'en tient par exemple à la définition de Simon.

Le traçage de modèle peut néanmoins être utilisé pour les parties bien définies de tâches mal définies s'il en existe [105]. Par ailleurs, il est aussi possible d'utiliser une approche itérative pour construire un modèle de tâche mal définie [130]. Cela peut se faire en partant d'un modèle de tâche grossier établi par des experts du domaine, puis à chaque itération, évaluer le modèle empiriquement pour vérifier s'il prédit correctement le comportement des utilisateurs et l'ajuster jusqu'à ce qu'un modèle satisfaisant soit trouvé. Cette stratégie a été utilisée, par exemple, pour construire un STI pour apprendre à des étudiants non francophones la distinction des temps de verbes en français [130].

1.3.2 La modélisation à base de contraintes

La seconde approche est la modélisation à base de contraintes (MBC) [115, 114]. Elle consiste à spécifier un ensemble de contraintes sur ce qu'est un comportement ou une solution correcte plutôt que de fournir un modèle de tâche explicite. Pendant une tâche, lorsqu'un apprenant viole une contrainte, un STI à base de contraintes peut diagnostiquer qu'une erreur a été faite, et peut fournir une aide sur la contrainte violée.

Contrairement au traçage de modèle, l'utilisation de contraintes ne permet pas de supporter des services tutoriels comme la génération de démonstrations ou la suggestion des prochaines actions à effectuer pour résoudre un problème. Ceci est un des principaux désavantages de cette approche. La MBC est recommandée pour les domaines où il est souhaité d'évaluer des états ou solutions, peu importe la stratégie utilisée pour produire les solutions (elle peut être appliquée pour des domaines où il n'y a pas de stratégie claire). Par contre, pour appliquer la MBC, il est nécessaire de définir des contraintes caractérisant les bons états/solutions pour une tâche, mais concevoir et sélectionner un ensemble de contraintes n'est pas toujours facile. De plus, les états/solutions pour certaines tâches mal définies ne sont pas assez informatifs pour permettre de spécifier un ensemble de contraintes pertinentes, et un très grand nombre de contraintes peut être nécessaire pour caractériser les états/solutions lorsqu'ils sont nombreux et largement

différents [95]. Une autre limite de l'utilisation de contraintes est que le cheminement menant à la violation d'une contrainte n'est pas pris en compte. Ceci peut avoir deux mauvaises implications. Premièrement, l'aide offerte par un STI peut être inadéquate, en particulier si le cheminement d'un apprenant diffère largement des solutions idéales [164, p.98]. Deuxièmement, si le raisonnement menant à une solution n'est pas évalué, il peut être impossible de distinguer entre une réponse correcte obtenue par chance ou obtenue par un raisonnement correct.

Malgré ces limites, la MBC a été appliquée avec succès pour plusieurs domaines mal définis et en particulier pour des tâches de conception et de résolution de problèmes (voir [114] pour un survol des applications). Un exemple de STI utilisant des contraintes est KERMIT/EER-Tutor, un STI pour l'apprentissage de la conception de diagrammes entité-relation [114]. Concevoir de tel diagrammes est une tâche mal définie parce qu'elle demande de la créativité (c'est une tâche de conception), il n'y a pas de stratégie claire pour l'effectuer, les définitions des problèmes sont généralement ambiguës et pour un problème, il peut y avoir plusieurs solutions discutables. Pour appliquer la MBC dans KERMIT, environ 100 contraintes ont été définies et l'interface de KERMIT a été conçue de façon à forcer les apprenants à utiliser les termes présents dans les définitions des problèmes pour nommer les éléments d'un diagramme. Cette restriction réduit grandement l'espace des solutions possibles. Cela illustre aussi une stratégie populaire pour construire des STI pour les domaines mal définis en général, qui est d'imposer une structure à un domaine pour le transformer en une forme mieux définie. Par contre, une limite de cette stratégie sur le plan pédagogique est que les apprenants devront éventuellement apprendre à effectuer les tâches dans des environnements moins structurés [119].

1.3.3 L'incorporation d'un système expert

La troisième approche qui est aussi la plus ancienne, consiste à intégrer un système expert comme module expert du STI [76, 93, 119, 32]. Le terme système expert est ici utilisé au un sens large qui inclut les systèmes à base de règles, les réseaux neuronaux, les systèmes de raisonnement à base de cas, etc. Pour certains domaines, utiliser un

système expert est une meilleure solution que l'utilisation du traçage de modèle ou de la MBC, car ces derniers sont des approches générales qui ne s'appliquent pas bien à tous les domaines (par exemple, pour le traitement du langage naturel, un système expert à base de méthodes statistiques est parfois plus appropriés que le traçage de modèle ou la MBC). L'utilisation des systèmes experts permet d'assurer deux fonctions essentielles d'un STI : générer des solutions expertes et valider les solutions de l'utilisateur (par comparaison avec les solutions de l'apprenant). Ainsi, un STI basé sur cette approche peut comparer ces solutions avec celles des apprenants, les utiliser comme démonstrations ou pour suggérer les prochaines étapes aux apprenants. Cette approche a été utilisée dans plusieurs STI tels que :

- GUIDON [32] est un STI pour apprendre à diagnostiquer les maladies infectieuses à partir des historiques de tests cliniques effectués sur des patients. GUIDON utilise MYCIN, un système expert à base de règles contenant environ 500 règles pour générer des solutions expertes. Ce qui rend MYCIN particulièrement approprié pour être utilisé dans un STI est que les règles de MYCIN ont une signification pour les humains. GUIDON utilise cette propriété pour présenter aux apprenants les règles qui réfutent leur raisonnement.
- AutoTutor [76] est un STI appliqué à plusieurs domaines dont la physique newtonienne et les concepts de base de l'informatique. AutoTutor enseigne au moyen de conversations en langage naturel avec les apprenants. Pour évaluer les réponses en langage naturel, AutoTutor utilise l'analyse sémantique latente pour calculer une distance sémantique avec des réponses attendues. L'*analyse sémantique latente* est une technique « boîte noire » dans le sens où elle est incapable de fournir une justification de son « raisonnement », contrairement à des systèmes experts comme MYCIN. Malgré cette limitation, elle est très efficace pour évaluer les réponses en langage naturel.
- DesignFirst-ITS [119] est un STI assistant des apprenants dans la conception de diagrammes UML à partir de descriptions textuelles. La construction de diagrammes UML est une tâche mal définie parce que c'est une tâche de conception, il y a un très grand nombre de possibilités, il n'y pas de solution unique, les énoncés des problèmes sont ambigus, et cela implique l'utilisation du langage naturel [119]. Une originalité

de DesignFirst-ITS est qu'il offre un environnement pratiquement sans limite aux apprenants contrairement à d'autres STI pour UML tel que Collect-UML, qui est basé sur la MBC [119]. DesignFirst-ITS évalue la construction étape par étape des diagrammes UML en les comparant avec des gabarits de solutions acceptables. Le processus d'appariement n'est pas trivial : il implique la recherche de synonymes et d'erreurs orthographiques, l'analyse des adjectifs, etc.

- Dans CanadarmTutor, un STI pour l'apprentissage de l'opération d'un bras robotisé, un système expert en planification de chemin est utilisé [93]. Ce STI est décrit plus loin dans ce document.

Dans tous ces cas, l'incorporation d'un système expert permet d'offrir des services tutoriels riches qui auraient été difficiles d'offrir avec le traçage de modèle ou la modélisation à base de contraintes. Néanmoins, les limites de l'utilisation d'un système expert sont que : (1) l'expertise du domaine doit être généralement explicitement encodée dans le système, ce qui est difficile lorsqu'elle est mal définie ; (2) dans certains cas (comme dans la planification de chemins), les inférences et solutions suggérées peuvent ne pas forcément être cognitivement plausible d'un point de vue de l'humain ; (3) la valeur pédagogique des explications générées peut être discutable, car elles consistent généralement à afficher la trace des actions qui ont mené à la prise de décision ; (4) la construction d'un système expert efficace nécessitera une tâche importante et coûteuse liée à l'acquisition des connaissances.

1.4 Les modèles pédagogiques pour offrir des services tutoriels dans les domaines mal définis

La section précédente a décrit les trois grandes approches pour offrir des services tutoriels et leurs limites pour les domaines mal définis. Cette description est seulement un aperçu des techniques disponibles pour construire des STI. Par exemple, l'approche des systèmes experts pourrait sans aucun doute être divisée en plusieurs autres catégories. Cette section présente le problème de la construction des STI pour les domaines mal définis selon une perspective complémentaire qui est celle des modèles pédagogiques.

Choisir un modèle pédagogique approprié est une décision importante, car cela peut considérablement faciliter la conception d'un STI. Bien que les modèles pédagogiques ne soient pas le centre d'intérêt de cette thèse (ce sont plutôt les approches pour supporter les services tutoriels et l'acquisition des connaissances du domaine), les modèles pédagogiques sont discutés ici afin de compléter le tour d'horizon de la problématique des domaines mal définis et ses solutions, et ainsi mieux situer les contributions de la thèse dans cette problématique.

1.4.1 Structurer l'apprentissage autour de l'étude de cas

Un premier modèle pédagogique efficace pour les domaines mal définis où la théorie du domaine est partielle et où il est important de faire des analogies avec des cas passés (par exemple, le diagnostic médical et l'argumentation légale) est de structurer l'apprentissage autour de l'étude de cas.

Un exemple de STI utilisant ce modèle pédagogique est CATO [5]. L'objectif de CATO est d'enseigner l'argumentation avec des cas à des étudiants novices en droit, et plus particulièrement la distinction des cas qui consiste à démontrer qu'un cas est significativement différent d'un second pour suggérer qu'il doit être jugé différemment. Identifier des distinctions entre cas qui sont positives pour la personne invoquant cet argument (qui l'aideront à défendre sa cause) est une tâche difficile et mal définie (elle demande de comprendre le langage naturel et les solutions sont discutables) [16]. Pour offrir des services tutoriels, CATO utilise un système expert incorporant un ensemble de cas indexés par des facteurs légaux [16]. Ce dernier est aussi utilisé dans CATO-Dial [16], une variation de CATO engageant l'étudiant dans des arguments légaux dans une cour simulée. CATO et CATO-Dial implémentent des services tutoriels élaborés pour supporter l'apprentissage par cas, et en particulier la comparaison des cas. Néanmoins, cette approche n'est pas la seule. Une stratégie plus simple est de laisser l'apprenant analyser et pratiquer un cas à la fois, comme dans GUIDON [32].

1.4.2 Supporter la métacognition

Un deuxième modèle pédagogique qui a été employé avec succès pour construire des STI pour des domaines mal définis est d'offrir un support métacognitif aux apprenants tout en donnant un support limité au niveau des connaissances du domaine. Ce modèle pédagogique a le bénéfice de ne pas demander un modèle détaillé de l'expertise du domaine.

Un exemple de STI supportant la métacognition dans un domaine mal défini est le STI expérimental conçu par Walker et al. [160] pour l'acquisition de compétences interculturelles. Ce domaine est mal défini, car expliquer des comportements culturels implique l'interprétation d'événements et du langage, et bien que des règles existent, il n'y a pas de théorie formelle du domaine [160]. Le système de Walker et al. demande aux étudiants de visionner des segments d'un film français touchant le sujet de l'immigration, et ensuite de répondre à des questions, puis finalement de contribuer à une discussion dans un forum de discussion. Chaque fois que l'apprenant écrit un message pour le forum, le système utilise des algorithmes d'analyse de mots-clés pour l'évaluer selon cinq critères de qualité tels que le lien avec le sujet et la prise en compte de plusieurs points de vues. Le système utilise cette évaluation pour suggérer des modifications à l'apprenant. L'apprenant doit ensuite effectuer au moins une modification à son message avant de le publier. Bien que le système ne possède pas de modèle de ce que sont les compétences interculturelles, il favorise l'apprentissage en promouvant un bon comportement d'apprentissage (écrire des messages de qualité) [160]. Le système de Walker et al. utilise un système expert pour supporter la métacognition, mais le traçage de modèle ou la MBC [114] peuvent également être utilisés pour supporter la métacognition.

1.4.3 Supporter l'apprentissage par investigation

Un troisième modèle pédagogique utilisé pour les domaines mal définis est l'apprentissage par investigation, une approche constructiviste [164, p. 312]. L'apprentissage par investigation consiste à découvrir des connaissances en accumulant des évidences et en

construisant et testant des hypothèses. Le rôle d'un STI pour l'apprentissage par investigation est de supporter le processus d'investigation plutôt que de fournir des connaissances aux apprenants. Par conséquent, il n'est pas nécessaire de définir un modèle de connaissances du domaine. Pour cette raison, l'apprentissage par investigation est un modèle pédagogique approprié pour les domaines mal définis où il n'y a pas de théorie complète du domaine. Plusieurs types de support peuvent être offerts par un STI pour l'apprentissage par investigation, incluant de contrôler le processus d'investigation, de fournir de l'aide au niveau métacognitif, et d'évaluer le raisonnement de l'apprenant et lui donner des conseils.

Un exemple de STI pour l'apprentissage par investigation est Rashi [39], un STI multidomaines qui a été appliqué à plusieurs domaines incluant la foresterie, la biologie, l'histoire et la géologie. Rashi présente des cas aux apprenants et les laisse recueillir des évidences, formuler des hypothèses et construire des argumentations pour vérifier les vérifier. Un apprenant peut recueillir des évidences avec des outils comme un outil d'entrevue et un explorateur d'images. Rashi supporte l'apprentissage à un niveau métacognitif par exemple en encourageant la considération de plusieurs hypothèses et la construction d'arguments ascendants ou descendants. Rashi possède aussi un modèle du domaine limité qu'il utilise pour détecter des relations manquantes dans les arguments et offrir des suggestions.

Une propriété intéressante de Rashi pour les domaines mal définis est qu'il permet à l'apprenant d'entrer des évidences additionnelles qui ne sont pas prédéfinies dans le modèle du domaine pour supporter les arguments. L'importance de construire des STI qui considèrent les connaissances générales d'un apprenant a aussi été noté par Easterday [44] pour le domaine de l'évaluation des politiques publiques. Ce domaine consiste à évaluer la probabilité qu'une politique mènera à un objectif [44]. Les problèmes d'évaluation de politiques sont mal définis puisqu'il n'y a pas de bonnes réponses et qu'il n'y a pas de stratégies faisant l'unanimité pour représenter les problèmes [44]. Easterday et al. ont observé que les novices et les experts utilisent parfois leurs connaissances générales pour discréditer des évidences et spéculer sur les conséquences d'une politique. Pour gérer ce

type d'erreurs, Easterday et al. proposent de (1) demander aux apprenants de raisonner avec les évidences et avec leurs connaissances générales séparément, puis de raisonner avec les deux, (2) de concevoir des interfaces utilisateurs qui permettent aux utilisateurs d'entrer leurs connaissances générales, et (3) de demander aux apprenants d'explicitier leur raisonnement et d'identifier où les connaissances générales interviennent. Récemment, ces propositions ont été prises en compte par Easterday et al. pour proposer le STI PolicyWorld [43].

1.4.4 Utiliser des histoires interactives

Un quatrième modèle pédagogique qui a été utilisé dans les STI pour les domaines mal définis est d'offrir des histoires interactives. Un système offrant des histoires interactives place l'apprenant au centre d'histoires où il peut influencer la direction ou le dénouement par la prise de décisions [88]. Cette approche a été démontrée dans AEINS [88], un STI pour apprendre l'éthique à travers des dilemmes moraux. AEINS offre de prendre part à des histoires interactives où l'apprenant fait face à de tels dilemmes. Prendre des décisions dans des dilemmes moraux est une tâche mal définie puisqu'il n'y a pas de bonne réponse. Le jugement de ce qui est bon ou mauvais est laissé à l'apprenant. AEINS donne la liberté d'agir pour laisser l'apprenant apprendre les conséquences de ses décisions. Pendant des dilemmes moraux, AEINS utilise la méthode socratique pour conduire un dialogue qui permet à l'apprenant de réfléchir sur les implications de ses décisions. AEINS a la capacité de générer et adapter des histoires interactives au profil de chaque apprenant. AEINS utilise le traçage de modèle pour mettre à jour ce modèle.

Apprendre avec des histoires interactives est similaire à l'apprentissage par investigation puisque c'est aussi une forme d'apprentissage par découverte, mais c'est différent puisque l'apprenant n'a pas besoin de recueillir des données, et construire et tester des hypothèses de façon systématique.

1.4.5 Structurer l'apprentissage autour de la collaboration

Un cinquième modèle pédagogique utilisé pour les domaines mal définis est de structurer l'apprentissage autour de la collaboration. L'objectif est de faire apprendre les participants en travaillant et en échangeant avec des pairs. Le bénéfice de ce modèle pour les domaines mal définis est que supporter la collaboration peut remplacer le besoin de construire un modèle du domaine. Un STI supportant la collaboration peut offrir différents niveaux de contrôle sur les interactions. Il peut guider la collaboration, offrir des conseils pour améliorer les interactions ou simplement afficher des indicateurs aux participants et leur laisser prendre les décisions pour améliorer leur collaboration [142]. Pour les domaines mal définis, un exemple de système collaboratif est celui de Walker et al. [160]. Comme mentionné, il favorise l'apprentissage en aidant les apprenants à faire de bonnes contributions aux discussions. Une approche particulière à la collaboration qui a obtenu un certain succès est de créer des compagnons virtuels qui interagissent avec les apprenants [31]. Un apprenant peut être en compétition, collaborer ou même apprendre en enseignant à des compagnon(s) virtuel(s). Bien que construire un compagnon virtuel ne nécessite pas un modèle détaillé du domaine, cela peut être une tâche difficile [105].

1.4.6 Autres modèles pédagogiques

La liste de modèles pédagogiques présentée dans cette section n'est pas exhaustive et plusieurs variantes sont possibles. Cette section a délibérément présenté des modèles pédagogiques moins conventionnels pour illustrer une gamme plus riche de modèles, mais utiliser une approche plus classique comme présenter des problèmes à résoudre à un apprenant et le supporter dans la résolution des problèmes est aussi possible.

1.5 Le domaine d'application de la thèse

Le concept de STI et la problématique des domaines mal-définis ayant été présentés, cette section décrit le domaine d'application de cette thèse, la manipulation robotique, et explique en quoi ce domaine est mal défini. La section fait ressortir les deux éléments

de problématiques de la thèse discutés en introduction (l'importance de tenir compte du raisonnement spatial et celle d'automatiser l'acquisition des connaissances du domaine). À noter que bien que les solutions proposées dans cette thèse soient appliquées à la manipulation robotique, elles ne sont pas limitées à ce domaine.

1.5.1 La manipulation robotique

La manipulation robotique est un domaine important qui couvre de nombreux domaines comme la manipulation d'un robot pour l'assemblage de voitures ou pour la détection de mines antipersonnelles [121]. L'entraînement à la manipulation de robots est une tâche qui se distingue de domaines enseignés par certains STI comme les mathématiques ou la physique dans le sens où l'apprentissage dans un STI pour ce domaine peut être fait difficilement sans un environnement virtuel simulé reproduisant des situations réelles de manipulation. La raison est que dans le domaine de la manipulation robotique, la dimension spatiale joue un aspect crucial dans les tâches de manipulation, par exemple lorsque des mouvements précis doivent être effectués avec un robot ou que les déplacements d'un robot doivent être planifiés, ou que des obstacles doivent être évités. De plus, le succès d'une tâche ou sa bonne conduite est souvent déterminé par des critères spatiaux comme l'atteinte d'un emplacement et la manipulation ou la transformation d'objets de l'environnement. Il pourra également être souhaité qu'un opérateur de robot devienne familier avec un environnement d'opération du robot. Dans tous les cas, la dimension spatiale est très importante à l'accomplissement des tâches.

Hormis l'aspect spatial, un deuxième défi important dans le domaine de la manipulation robotique est l'explicitation du processus de décision des humains qui manipulent un robot. En effet, une des raisons pour laquelle les robots sont contrôlés par des humains dans plusieurs domaines est qu'il serait difficile de construire un robot entièrement autonome encodant le processus de décision et le jugement des opérateurs humains. Par exemple, cela est le cas du robot semi-autonome décrit par Nanayakkara et al. [121] où un humain influence le comportement d'un robot démineur, et du bras robotisé Canadarm2 présenté dans la prochaine sous-section. Dans de tel cas, la manipulation robotique est

un domaine mal défini dans le sens où les processus de décision de l'opérateur sont difficilement formalisables (selon la définition de Simon [141], un domaine pour lequel il y a des problèmes n'ayant pas de stratégies claires pour obtenir une solution sont dits mal-définis).

1.5.2 CanadarmTutor, un STI pour la manipulation de Canadarm2

Les deux défis mentionnés dans les paragraphes précédents se présentent aussi pour le domaine de la manipulation du bras robotisé Canadarm2, qui est le domaine d'application plus spécifique de la thèse. Cette sous-section présente ce domaine par le biais de la présentation de CanadarmTutor, un STI pour ce domaine dans lequel les propositions de cette thèse sont appliquées. Les informations présentées sur la manipulation de Canadarm2 dans la thèse qui ont permis la réalisation du travail de cette thèse avec CanadarmTutor sont des entrevues avec Daniel Dubois, un membre du laboratoire GDAC qui a observé l'entraînement d'astronautes à l'Agence Spatiale Canadienne de Saint-Hubert (Canada) en mars 2006, la lecture de son rapport [40] et la lecture des rapports de David Cournoyer, un étudiant qui a assisté à l'entraînement d'astronautes en 2003 [34, 35].

CanadarmTutor [93] est un prototype développé par les laboratoires de recherche GDAC de l'Université du Québec à Montréal et Planart de l'Université de Sherbrooke, pour l'entraînement à la manipulation de Canadarm2 (cf. fig. 1.3), un bras robotisé à 7 degrés de liberté installé sur la station spatiale internationale (SSI). CanadarmTutor offre à l'utilisateur de manipuler le bras dans un environnement virtuel simulé. L'interface de CanadarmTutor (cf. figure 1.4) reproduit partiellement le panneau de contrôle utilisé par les astronautes (cf. figure 1.5). Les boutons et molettes de l'interface permettent à l'utilisateur d'associer les caméras à chaque moniteur et d'ajuster le zoom, la rotation horizontale et la rotation verticale des caméras sélectionnées. Le bras canadien est contrôlé par le clavier en mode cinématique directe. Les champs du bas de l'interface affichent les actions de l'apprenant et l'état du simulateur. Les menus de l'interface permettent de définir les préférences, de choisir un programme d'apprentissage et de demander une

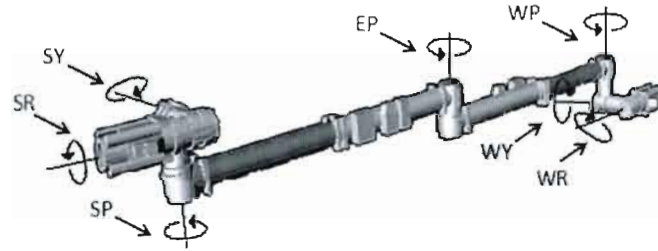


Figure 1.3 Le bras robotisé Canadarm2

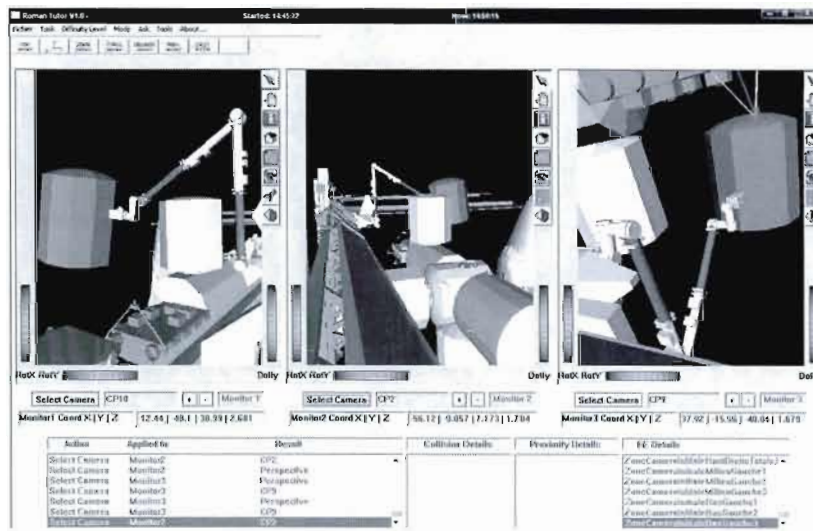


Figure 1.4 CanadarmTutor

aide ou des démonstrations.

1.5.3 Comment donner une expertise du domaine à CanadarmTutor ?

La tâche d'intérêt dans CanadarmTutor est de faire passer le bras robotisé d'une configuration à une autre. Pour fournir une expertise du domaine à CanadarmTutor, l'approche utilisée initialement a été d'intégrer un système expert sous la forme d'un planificateur [19, 93, 124]. Ce planificateur peut générer une trajectoire évitant les obstacles entre n'importe quelles configurations du bras canadien. Le planificateur permet aussi de suivre les solutions de l'apprenant. Néanmoins, les trajectoires générées ne sont pas



Figure 1.5 Un astronaute opérant Canadarm2 (NASA)

toujours réalistes ou faciles à suivre, parce qu'elles ne sont pas basées sur l'expérience humaine, et qu'elles ne couvrent pas d'autres aspects importants de la manipulation comme le choix des caméras et l'ajustement de leurs paramètres. Aussi, le planificateur ne supporte pas des services tutoriels importants comme l'estimation des connaissances manquantes ou erronées des apprenants.

Les deux autres grandes approches d'acquisition et de représentation des connaissances (modélisation à base de contraintes et l'analyse cognitive de tâche) auraient aussi un succès limité dans CanadarmTutor puisque ce dernier opère dans un domaine mal défini (au sens de Simon [141]). En effet, pour passer d'une configuration du bras à une autre il n'existe pas de façon simple d'explicitier comment déterminer les meilleures opérations à partir d'une configuration donnée pour atteindre un objectif. La raison est que la meilleure séquence de mouvements n'est pas nécessairement la plus courte, mais est la plus sécuritaire au sens de l'opérateur, un critère qui est subjectif et qui dépend de nombreux facteurs comme le degré de familiarité avec les différentes séquences de mouvements, avec les caméras et les zones de l'environnement. Ainsi, une séquence est acceptable si elle respecte le protocole de sécurité, qu'elle est accomplie sans fautes et que l'opérateur la conçoit comme étant sécuritaire en termes de ses capacités [40]. Ces

critères ne peuvent être capturés que partiellement sous forme de règles et de contraintes. Cela illustre la nécessité de proposer de nouvelles approches d’acquisition et de représentation des connaissances pour faire face à la difficulté de formaliser de tels domaines procéduraux et mal définis.

1.5.4 L’importance du raisonnement spatial pour la manipulation de Canadarm2

De plus, les modèles de représentation de connaissances pour les STI axés sur la résolution de problème ne tiennent pas compte de la dimension spatiale présente dans les domaines comme la manipulation de Canadarm2. Dans ce dernier cas, la cognition spatiale est un aspect crucial. En effet, les connaissances et habiletés spatiales pour la manipulation de Canadarm2 sont variées et incluent par exemple de connaître les meilleures caméras pour effectuer des opérations dans certaines zones de l’environnement, de savoir planifier des séquences de mouvements pour déplacer le bras robotisé, d’estimer les distances, de connaître les modules de la station spatiales, etc. Pour offrir un entraînement de qualité, CanadarmTutor devrait donc non seulement être capable de vérifier que les opérations effectuées par un apprenant ne sont pas dangereuses (par exemple, qu’elles n’empruntent pas un passage étroit ou qu’elles ne résultent pas en une grande proximité avec un obstacle) et qu’elles respectent le protocole de sécurité, mais aussi d’offrir à l’apprenant des situations personnalisées pour l’amener à développer ses capacités en termes de connaissances et de raisonnement spatial dans l’environnement de Canadarm2.

Les deux grands défis que sont la difficulté de formaliser la tâche et comment offrir des services tutoriels pour le raisonnement spatial seront abordés en détails dans les chapitres 2, 3, 4 et 5.

1.6 Conclusion

Ce premier chapitre a décrit le concept de STI et présenté la problématique des domaines mal définis. La section 1.2 en a clarifié la définition. Puis les sections 1.3 et 1.4 ont abordé

respectivement les avantages et limites des approches classiques pour offrir des services tutoriels dans ces domaines, et le choix d'un modèle pédagogique pour ces domaines. Finalement, la section 1.5 a décrit comment les deux défis identifiés dans l'introduction (l'importance de tenir compte du raisonnement spatial et celle d'automatiser l'acquisition des connaissances du domaine) se présentaient dans le domaine d'application de la thèse qu'est la manipulation robotique. En particulier, la section a présenté Canadarm-Tutor, le cadre d'application de la thèse. Le prochain chapitre décrit le modèle cognitif développé pour supporter le raisonnement spatial dans les STI et son application dans CanadarmTutor.

CHAPITRE II

UN MODÈLE COGNITIF POUR ÉVALUER LE RAISONNEMENT SPATIAL

De nombreuses activités comme conduire une voiture impliquent l'utilisation de représentations, de compétences et d'habiletés spatiales. Les *compétences spatiales* (spatial abilities) et les *habiletés spatiales* (spatial skills) forment les deux extrémités d'un continuum. Les premières sont plus générales et peuvent être appliquées dans différents domaines, comme la rotation mentale d'objets ou le changement de perspectives (voir par exemple, Hegarty et Waller [85], et Kozhevnikov et Hegarty [98]). Les secondes réfèrent à des habiletés plus spécialisées qui impliquent la manipulation correcte de représentations spatiales pour accomplir des buts particuliers.

Bien que les habiletés et représentations spatiales soient généralement acquises par la pratique, elles peuvent être réifiées sous forme de connaissances déclaratives qui peuvent être explicitement manipulées dans le raisonnement ou communiquées sous forme de descriptions comme des cartes routières ou des instructions de navigation [156]. L'apprentissage de telles habiletés ou représentations est souvent une tâche coûteuse en temps. Pour réduire les coûts d'entraînement et reproduire des scénarios complexes, l'entraînement peut être effectué avec des simulations. À cet effet, des environnements réalistes et immersifs ont été développés ; par exemple pour l'entraînement de soldats à des tâches coordonnées de groupes [153]. Néanmoins, l'entraînement à base de simulation échoue à offrir un apprentissage optimal, puisque les scénarios d'entraînement et les rétroactions ne sont pas personnalisés aux représentations, habiletés et compétences spatiales des

apprenants.

Il est possible d'améliorer cette situation de deux façons. Premièrement, des experts observent des apprenants pendant leur entraînement pour intervenir dans une session d'apprentissage et créer des activités personnalisées. Deuxièmement, un système d'entraînement augmenté de techniques dérivées du domaine des STI évalue les représentations et habiletés spatiales et génère des scénarios et une assistance personnalisée.

Bien que la recherche sur les STI a débutée il y a plus de deux décennies, le problème de construire des systèmes d'entraînement à base de simulation qui fournissent une assistance personnalisée aux représentations, compétences ou habiletés spatiales d'un apprenant est un problème qui n'a pas été encore abordé par les chercheurs. Or, de nombreuses tâches comportent une dimension spatiale considérable et en tenir compte favoriserait l'apprentissage. En fait, deux facteurs sont importants et doivent être pris en compte pour développer un modèle cognitif de tâche qui considère les représentations et les habiletés spatiales : la nature des représentations spatiales et leur modélisation. Deux questions importantes pour la recherche se posent : quelles représentations spatiales et quels modèles informatiques choisir pour ces représentations ? Les deux prochaines sections apportent des éléments de réponses à ces questions

2.1 Quelle est la nature des représentations spatiales ?

Cette question est étudiée depuis plus de 50 ans [154]. De façon générale, deux grand types de représentations sont proposés : les *cartes mentales* aussi appelées *représentations allocentriques* et les *représentations égocentriques* [120].

Tollman [154] a initialement proposé le concept de cartes mentales, après avoir observé le comportement de rats dans des labyrinthes. Il a postulé que les rats construisent et utilisent des cartes mentales de l'environnement pour prendre des décisions spatiales. O'Keefe et Nadel [131] ont recueilli des évidences neurologiques des cartes mentales. Ils ont entre autres observé que certaines cellules nerveuses des rats (appelées « cellules lieu ») s'activent de façon similaire quand un rat est dans la même position spatiale,

peu importe ce qu'il fait. Sur la base de ces résultats chez les rats, O'Keefe et Nadel ont formulé l'hypothèse que les humains utilisent non seulement des représentations spatiales égocentriques, qui encodent l'espace selon la perspective d'une personne, mais aussi des cartes mentales allocentriques, indépendantes de tout point de vue.

Selon O'Keefe et Nadel [131], une représentation égocentrique décrit la position d'un objet selon la perspective d'une personne. Pour être utile, une représentation égocentrique doit être constamment mise à jour, en fonction de la perception d'une personne. Ce processus appelé *intégration de déplacements* (path integration) est supporté par des évidences expérimentales solides [46, 116]. Par exemple, des chercheurs ont observé que des fourmis peuvent retourner directement à leur position initiale, même après avoir parcouru plus de 100 mètres sur un terrain plat, ce qui indique qu'ils ont constamment mis à jour leur représentation de la relation entre la position actuelle et la position initiale au cours de leur déplacement [46]. Dans le contexte de la navigation par instructions, les représentations égocentriques décrivent des repères relatifs le long d'un trajet, et la navigation consiste alors à effectuer les bons déplacements à partir de chaque point de repère pour atteindre le suivant [156]. Les connaissances égocentriques sont normalement acquises par l'expérience, mais elles peuvent aussi être acquises directement sous forme d'instructions textuelles, par exemple. La navigation par instructions est extrêmement inflexible et laisse peu de place à la déviation. En effet, choisir des directions par rapport à des points de repère dépend de la position relative d'une personne par rapport aux points de repère. Par conséquent, des déviations de trajectoire peuvent facilement perturber l'accomplissement d'une tâche de navigation. L'encodage ou le rappel incorrect peut aussi sérieusement compromettre l'accomplissement du but.

Selon Tversky [156], les représentations égocentriques peuvent être suffisantes pour effectuer des tâches comme naviguer dans un environnement, mais elles ne sont pas adéquates pour effectuer des raisonnements spatiaux complexes. Pour le raisonnement qui demande de l'inférence, les humains construiraient des cartes mentales qui ne contiendraient pas de mesures précises, mais plutôt les principales relations spatiales entre les objets de l'environnement. De telles représentations n'encodent pas une perspective particulière, mais

permettent d'adopter plusieurs perspectives. Les cartes mentales sont aussi sujettes à des erreurs d'encodage et de rappel. Par contre, récupérer d'une erreur est généralement plus facile en se basant sur une carte mentale que sur des représentations égocentriques. De plus, il est plus efficace de mettre à jour une représentation allocentrique, puisque seulement la position du soi change lors d'un déplacement (les relations entre les objets ne changent pas), contrairement aux représentations égocentriques qui demandent de modifier les relations spatiales de tous les objets par rapport à soi. Par ailleurs, certains chercheurs suggèrent qu'en plus des représentations allocentriques et égocentriques, les humains pourraient utiliser aussi des cadres de référence « semi-allocentrique » [111], où la position d'un objet est encodée selon un axe dominant présent dans l'environnement, comme celui formé par deux édifices distants, lorsqu'un tel axe est présent.

Récemment, des cellules lieu ont été découvertes dans l'hippocampe humain [45]. L'observation de cellules nerveuses chez les rats a aussi amené à la découverte de cellules sensibles à l'orientation de la tête [151, 152], des cellules lieu modulées par la vitesse de déplacement [17] et des cellules lieu s'activant périodiquement lorsqu'un rat se déplace en ligne droite [112]. À la lumière de telles recherches, et d'autres en neuroscience, neuropsychologie [21, 52, 140], psychologie et d'autres disciplines, il ne fait aucun doute que les humains utilisent des représentations spatiales égocentriques et allocentriques [24, 120].

2.2 Comment implémenter les représentations spatiales dans un modèle informatique, et dans un STI ?

La deuxième question importante qui doit être considérée pour développer un modèle cognitif dans un STI qui tient compte des représentations et habiletés spatiales est comment implémenter les représentations spatiales dans un modèle informatique. Une partie de la réponse à cette question peut être trouvée dans les modèles dérivés de la recherche expérimentale sur la cognition spatiale. Par contre, de tels modèles se spécialisent généralement sur des phénomènes particuliers comme la perception visuelle et la reconnaissance de mouvements [28], la navigation dans des environnements 3D [112, 84] et

l'imagerie mentale et l'inférence à partir de descriptions spatiales [26]. Les rares modèles qui tentent de donner une explication plus générale de la cognition spatiale n'ont aucune implémentation informatique ou seulement une implémentation partielle. Par exemple, le modèle de Gunzelmann et Lyon [77] ne possède qu'un modèle informatique partiel : plusieurs éléments cruciaux du modèle ne sont pas assez détaillés pour être implémentés. La plupart des modèles cognitifs de la cognition spatiale peuvent être catégorisés comme fournissant des structures à un niveau symbolique ou connexionniste. Les modèles basés sur les réseaux de neurones peuvent simuler avec un succès considérable des comportements de bas niveau comme le comportement d'activation des cellules sensibles à l'orientation de la tête [137] et l'intégration de déplacements avec des cartes mentales [112]. Alternativement et avec certaines particularités, les modèles symboliques qui utilisent des représentations allocentriques [28, 26, 77] représentent les relations spatiales comme des liens de type « a r b » où « r » représente une relation spatiale telle que « est à gauche de » ou « est par-dessus » et où « a » et « b » sont des représentations mentales d'objets. Chaque représentation allocentrique est encodée selon un cadre de référence implicite ou explicite comme un système de coordonnées. Cette représentation des cartes mentales est en accord avec les travaux de chercheurs comme Tversky [156], qui suggèrent que les cartes mentales sont encodées en mémoire sémantique. Par contre, une représentation égocentrique est généralement représentée par une relation de type « s r o » où « r » dénote une relation spatiale et « s » et « o » dénotent des représentations mentales du soi et d'un objet, respectivement.

Les modèles cognitifs de la cognition spatiale cités sont néanmoins limités de plusieurs façons pour être appliqués dans des STI et leur donner une capacité de tenir compte de la dimension spatiale des tâches. Tout d'abord, les modèles cognitifs qui expliquent des phénomènes spécifiques de la cognition spatiale, comme ceux de Carruth et al. [28] et Byrne et Johnson-Laird [26] ne permettent pas d'obtenir une compréhension globale de la performance cognitive. Pour y arriver, il faudrait plutôt utiliser des modèles basés sur une théorie unifiée de la cognition qui a une implémentation informatique. Or, les modèles actuels n'offrent pas d'implémentation informatique, par exemple : Gunzelmann

et Lyon [77]. Deuxièmement, les modèles connexionnistes ne sont pas appropriés pour être utilisés dans un STI, car, pour être enseignée, la connaissance doit être explicite et ils modélisent généralement des processus cognitifs de bas niveau. Finalement, une autre considération importante est qu'aucun des modèles mentionnés n'a été implémenté dans un STI pour favoriser l'acquisition des représentations et des habiletés spatiales. En fait, une telle implémentation exigerait des changements majeurs, puisque ce contexte comporte des défis particuliers. La principale difficulté consiste à développer des mécanismes pour évaluer les apprenants, en dépit de l'exiguïté du canal de communication entre l'apprenant et le système (généralement, un clavier, une souris et un moniteur) [163].

Pour ces raisons, il a été choisi pour cette thèse d'utiliser un modèle cognitif pour les STI qui est basé sur une théorie unifiée de la cognition, et de l'étendre sur la base des recherches sur la cognition spatiale.

2.3 Le modèle cognitif de Fournier-Viger et al.

Pour développer le modèle cognitif, un modèle cognitif symbolique basé sur une théorie unifiée de la cognition a été sélectionné comme fondement. Le choix s'est arrêté sur un modèle que l'auteur de ce document a développé durant sa maîtrise [54, 56, 57, 58]. Ce modèle est une simplification des théories cognitives ACT-R [12] et MIACE [109], conçue spécialement pour suivre le raisonnement des apprenants dans des STI. Ce modèle a été choisi, car il possède essentiellement les mêmes caractéristiques que le modèle des tuteurs cognitifs d'Anderson et al. [13] qui est la référence pour la construction de tuteurs cognitifs, et parce qu'il comporte en plus quelques originalités comme l'association d'ontologies aux connaissances sémantiques pour faciliter leur réutilisation entre domaines et permettre la création d'objets d'apprentissage (voir [57], pour des détails sur ces originalités). Les prochaines sous-sections présentent les aspects pertinents du modèle (pour plus de détails, le lecteur pourra se référer à [54] et la section suivante décrit son extension pour tenir compte de la dimension spatiale.

2.3.1 Le modèle théorique

Le modèle organise les connaissances en connaissances sémantiques et procédurales. Les connaissances sémantiques sont des connaissances descriptives qui ne sont pas associées aux événements [155]. Le modèle considère ces dernières comme des *concepts* au sens large. Selon des recherches récentes [78], un humain peut considérer jusqu'à quatre instances de concepts dans l'accomplissement d'une tâche, mais il peut en grouper plusieurs pour les traiter comme une seule instance [78]. Dans le modèle, ces connaissances sont nommées *concepts composés*, par opposition aux *concepts primitifs* qui sont syntaxiquement indécomposables. Par exemple, l'expression « PMA03 estConnectéSous Lab02 » est une représentation décomposable contenant trois symboles atomiques qui représente la connaissance que le module « PMA03 » de la SSI est connecté sous le module « LAB02 » de la SSI (en supposant l'axe de référence standard). De cette façon, la sémantique d'un concept composé est fonction de ses composantes. Alors que les concepts sont stockés dans la mémoire sémantique, les instances de concepts qui sont manipulées par les processus cognitifs apparaissent dans la mémoire de travail, et sont caractérisées par leur contexte mental et temporel [109]. Par conséquent, chaque occurrence d'un symbole tel que « LAB02 » est vue comme une instance distincte d'un même concept.

La mémoire procédurale encode des façons d'atteindre des buts (intentions) automatiquement en manipulant des connaissances sémantiques. Elle est constituée de *procédures* qui sont déclenchées une à la fois en fonction de l'état de l'architecture cognitive [12]. Contrairement aux connaissances sémantiques, l'activation d'une procédure ne requiert pas l'attention. Par exemple, une procédure peut être de sélectionner une caméra avec l'interface de CanadarmTutor. Comme MIACE [109], le modèle distingue les procédures primitives des procédures complexes. Alors qu'une procédure primitive est une action atomique, une procédure complexe instancie un ensemble de buts pouvant être ensuite atteints soit par une procédure complexe ou primitive.

Les *buts* sont des intentions qu'un humain peut avoir tel que le but d'additionner deux nombres [109]. À chaque moment, l'architecture cognitive a un but qui représente son

intention. Ce but est choisi parmi l'ensemble des buts actifs [14]. Il peut y avoir plusieurs façons correctes ou erronées (procédures) d'accomplir un but. Le modèle voit les buts comme un type particulier de concept composé instancié avec zéro ou plusieurs instances de concepts qui sont l'objet du but. Par exemple, une instance du concept « CUPOLA01 » pourrait être paramètre du but « ChoisirLaMeilleureCaméraPourVoirLeModule », ce qui représenterait l'intention de choisir la meilleure caméra pour voir le module « CUPOLA01 » de la SSI. Les paramètres d'un but peuvent restreindre les procédures qui peuvent être déclenchées pour le but. Les paramètres d'un but sont aussi un moyen de transmission de connaissances sémantiques entre la procédure complexe qui le crée et la procédure qui sera déclenchée pour l'accomplir.

En spécifiant les paramètres d'un but, une procédure détermine quelles procédures peuvent être déclenchées pour ce but, et permet à la prochaine procédure d'accéder à ces instances de concepts.

2.3.2 L'implémentation du modèle

Une structure informatique a été développée pour décrire les processus cognitifs pour une activité selon la théorie cognitive précédente [54]. Cette structure définit un ensemble d'attributs pour décrire les concepts, buts et procédures. La valeur d'un attribut peut être un ensemble de concepts, buts ou procédures, ou des données arbitraires comme des chaînes de caractères. La figure 2.1 illustre la liste des attributs correspondant à chaque type de connaissance. Ils sont identifiés par le symbole « + ».

Tout d'abord, les concepts primitifs ont deux attributs qui définissent respectivement leur super-concept(s) et sous-concept(s) en fonction d'une ontologie [54, 57]. Les concepts composés ont un attribut supplémentaire qui indique le type de concepts de chacune de leurs composantes. Les buts ont deux attributs. Le premier indique les types de concepts pris en paramètres par le but. Le second contient un ensemble de procédures qui peuvent être utilisées pour le but. Les procédures ont 5 attributs. « But » indique le but pour lequel la procédure a été définie. « Paramètres » spécifie les types des concepts pris en

Concept primitif	Concept composé	But	Procédure
super-concept(s) +	super-concept(s) +	Paramètres +	But +
sous-concept(s) +	sous-concept(s) +	Procédures +	paramètre(s) +
	Composantes +		méthode / script +
	général*		observable +
	Validité*		validité +
	composantesRetrait*		requêteDeRappel*
			appariementRappel*

+ = Attribut du modèle initial
 * = Attribut ajouté dans le modèle étendu

Figure 2.1 Attributs du modèle cognitif

arguments. Pour chaque procédure primitive, « Méthode » pointe à une méthode Java qui exécute une action atomique, et l'attribut « Observable » spécifie si la procédure correspond à une action observable ou si elle est plutôt une étape mentale. Pour les procédures complexes, « Script » indique un ensemble de sous-buts à réaliser avec zéro ou plusieurs contraintes sur l'ordre d'accomplissement. Finalement, « Validité » indique si une procédure est valide.

Un certain nombre d'outils informatiques ont été développés pour travailler avec ce modèle comme un système auteur graphique pour faciliter l'entrée des connaissances. Le lecteur pourra se référer à [54] pour une description détaillée de ces outils. Les outils pertinents pour cette thèse, nommés *l'interpréteur* et le *traceur de modèle*, seront décrits à la section 2.5. Le modèle informatique a été employé dans des prototypes élémentaires de STI pour plusieurs domaines comme la réduction booléenne [54], le génie génétique et l'analyse de circuits [1].

2.4 L'extension du modèle cognitif

Bien que le modèle se soit montré utile pour offrir une assistance personnalisée dans des STI [54, 57], son implémentation met l'accent sur l'acquisition des connaissances procédurales. La raison est que l'implémentation n'incorpore pas un élément clé de plusieurs théories cognitives qu'est le rappel de connaissances sémantiques (le modèle cognitif im-

plémenté est une simplification des architectures cognitives ACT-R et MIACE qui ont cette capacité). Ainsi, il est impossible de spécifier que pour atteindre un but, l'un doit se rappeler du concept « CameraCP5 estAttachéeÀ S1P1TrussLeft01 » (la caméra CP5 est attachée au module S1P1TrussLeft01 de la SSI), et le manipuler avec une procédure ensuite. Or, évaluer les connaissances sémantiques est essentiel pour évaluer les représentations spatiales, si l'on considère que les cartes mentales sont encodées en mémoire sémantique [156].

Pour pallier cette limite, le modèle a été étendu [60, 61, 62, 63]. Pour modéliser le rappel des connaissances sémantiques et ainsi les connecter aux connaissances procédurales, l'extension du modèle ajoute un mécanisme de rappel. Son fonctionnement est inspiré de celui de la théorie cognitive ACT-R [12], dans le sens qu'une procédure (l'équivalent d'une règle dans ACT-R) peut demander le rappel d'une instance de concept en spécifiant certains critères de rappel, une procédure peut s'exécuter conditionnellement au résultat d'un rappel, le résultat d'un rappel est stocké dans un tampon ne contenant que le dernier élément rappelé et que les procédures peuvent accéder à ce tampon pour effectuer leur opération. Toutefois, le mécanisme proposé est une simplification importante du mécanisme de rappel d'ACT-R, car il n'y a pas de calcul d'activation pour déterminer quelle connaissance est rappelée. La raison est que le but du modèle présenté dans ce chapitre est de pouvoir suivre le raisonnement de l'apprenant et non pas d'effectuer des simulations cognitives comme cela est fait avec ACT-R (l'utilisation du modèle sera décrite à la section 2.5). Le mécanisme d'ACT-R a été choisi comme inspiration pour étendre le modèle cognitif, car ce dernier était déjà basé sur cette théorie.

L'addition du mécanisme de rappel est faite ainsi. Une procédure peut dorénavant demander le rappel d'une instance de concept composé en spécifiant une ou plusieurs contraintes sur les valeurs de ses composantes. Ceci est accompli par l'ajout d'un nouvel attribut nommé « requêteDeRappel ». Il spécifie l'identificateur d'un concept composé à être rappelé et zéro ou plusieurs restrictions sur la valeur de ses composantes. À titre d'exemple, la table 2.3 illustre la procédure « RappelCaméraPourVueModuleSSI ». L'exécution de cette procédure demande le rappel de la connaissance d'une caméra qui

donne une vue du module de la SSI pris en paramètre par la procédure. L'attribut « requêteDeRappel » spécifie qu'une instance de concept de type « RelationSpatialeCameraEtModuleSSI » doit être rappelée (une relation indiquant qu'une caméra donne une vue d'un module de la SSI) et que sa première composante doit être de type « Caméra » et que sa deuxième composante doit être de type « ModuleSSI » et s'apparier avec le paramètre de la procédure. Un rappel réussi suivant l'exécution de cette procédure résultera en le dépôt d'une instance de « RelationSpatialeCameraEtModuleSSI » dans un tampon spécial qui accepte la dernière instance rappelée (le tampon ne conserve que la dernière instance rappelée). Ensuite, la prochaine procédure exécutée peut accéder à cette instance de concept dans le tampon pour accomplir son but. Dans le modèle étendu, grâce à l'attribut « AppariementRappel », une procédure peut être spécifiée comme étant seulement exécutable lorsque le tampon de rappel contient une instance de concept d'un certain type.

Dans le modèle étendu, toutes les connaissances sémantiques ne peuvent pas être rappelées. Une distinction est faite sur une base pédagogique entre *connaissances sémantiques générales* et *contextuelles*. Les connaissances générales sont des connaissances sémantiques toujours vraies ou toujours fausses (pour encoder des conceptions erronées courantes) pour un curriculum. Une telle connaissance est que la taille de l'extrémité de Canadarm2 mesure environ 1 m. Les connaissances générales sont des concepts composés, car pour être utiles, elles doivent représenter des relations. Pour être utilisée correctement, une connaissance générale doit être (1) acquise correctement, (2) être rappelée correctement et (3) être manipulée par des connaissances procédurales valides. Quand une connaissance générale est rappelée, elle est instanciée avec des composantes prédéterminées qui sont toujours les mêmes (une connaissance générale peut être vue comme une instance de concept composé statique, stockée en mémoire à long terme). En revanche, les connaissances contextuelles sont les connaissances obtenues par l'interprétation d'une situation. Ce sont des instances de concepts qui sont dynamiquement instanciées lors d'une situation. Par exemple, l'information que la valeur de rotation du joint « WY » de Canadarm2 est 42° est une connaissance contextuelle acquise par la

lecture du tableau de bord. Dans le modèle étendu, seules les connaissances générales peuvent être rappelées, car ce sont les connaissances qu'il est souhaitable d'évaluer sur un plan pédagogique pour voir si un apprenant possède les connaissances sémantiques prescrites par un curriculum. Pour les connaissances contextuelles, le modèle permet la définition de procédures primitives qui simule l'acquisition par la perception visuelle. Par exemple, une procédure primitive peut être définie pour lire la valeur de rotation d'un joint de l'interface utilisateur et créer une instance de concept composé correspondante.

Pour inclure la distinction entre connaissances générales et contextuelles dans le modèle informatique, trois attributs ont été ajoutés aux concepts composés. « général » indique si un concept est général. Pour les concepts généraux, « validité » indique si le concept est vrai ou faux et optionnellement l'identifiant d'un concept vrai correspondant. De plus, l'attribut « composantesRetrait » spécifie pour chaque concept général, l'ensemble des concepts à instancier comme composantes quand le concept est rappelé. La table 2.1 présente un exemple de concept encodant la connaissance que le module « MPLM02 » peut être vu par la caméra « CP7 ». L'attribut « validité » révèle que cette connaissance est erronée et que l'information valide équivalente est le concept « CP7_vue_de_MPLM01 » (cf. table 2.2). De plus, l'attribut « super-concept(s) » indique que ces deux concepts sont des spécialisations de « RelationSpatialeCameraEtModuleSSI », le concept d'une relation de vision entre une caméra et un module de la SSI. La connaissance fausse « MPLM02 » a été encodée pour représenter une erreur courante qui est de confondre le module « MPLM02 » avec le module « MPLM01 » pour le choix de la caméra « CP7 ».

Sur la figure 2.1 les attributs ajoutés pour le modèle étendu sont indiqués par le symbole « * ».

2.5 Deux outils pour exploiter les aspects dynamiques du modèle

Pour simuler les aspects dynamiques du modèle cognitif, deux outils ont été développés [54, 62]. Ils se nomment « traceur de modèle » et « interpréteur » et sont la fondation des services tutoriaux que le modèle peut offrir.

Attributs	Valeurs
super-concept(s)	RelationSpatialeCameraEtModuleSSI
composantes	Camera, ModuleSSI
composantesRetrait	CP7, MPLM02
général	True
validité	False (valide=CP7_vue_de_MPLM02)

Tableau 2.1 Définition partielle du concept « CP7_vue_de_MPLM02 »

Attributs	Valeurs
super-concept(s)	RelationSpatialeCameraEtModuleSSI
composantes	Camera, ModuleSSI
composantesRetrait	CP7, MPLM01
général	True
validité	True

Tableau 2.2 Définition partielle du concept « CP7_vue_de_MPLM01 »

Attributs	Valeurs
but	ButRappelerCaméraPourVueModuleSSI
paramètres	(ModuleSSI : m)
requêteDeRappel	ID : RelationSpatialeCameraEtModuleSSI, Arg1 :Camera, Arg2 :ModuleSSI :m
validité	True

Tableau 2.3 Définition partielle de la procédure « RappelCaméraPourVueModuleSSI »

Le rôle de l'interpréteur est de simuler un comportement décrit avec le modèle cognitif. Il est utilisé par le traceur de modèle. L'interpréteur possède la capacité de sauvegarder son état pour être arrêté et redémarré ultérieurement (l'utilité sera présentée plus loin). Un état est défini comme l'ensemble des but(s) actifs. Au départ, il contient un but (celui d'accomplir un exercice). À chaque cycle, l'interpréteur demande au traceur de modèle de choisir le but actif parmi l'ensemble des buts instanciés. L'interpréteur détermine ensuite quelles procédures peuvent être exécutées pour le but. Le traceur de modèle doit choisir une procédure pour poursuivre la simulation. Pour une procédure primitive, l'interpréteur calcule le résultat, et enlève le but de l'ensemble de buts. Pour l'exécution d'une procédure complexe, l'interpréteur instancie les sous-buts de la procédure et les ajoute à l'ensemble des buts instanciés. Un but réalisé par une procédure complexe est retiré de l'ensemble de buts lorsque tous ses sous-buts sont réalisés. Si l'exécution d'une procédure demande le rappel d'une connaissance sémantique générale, et que plusieurs choix sont possibles, l'interpréteur demande au traceur de modèle de choisir la connaissance à être rappelée (la façon dont ce choix est réalisé est décrit dans la partie de cette section décrivant le traceur de modèle). L'interpréteur arrête quand il ne reste aucun but.

Un auteur doit spécifier un but principal pour chaque exercice de résolution de problème. La simulation de résolution avec l'interpréteur génère une structure telle que celle montrée à la figure 2.2.A. Sur cette figure le but principal *G* a été accompli par une procédure complexe *CP1* qui a instancié trois sous-buts. Alors que le premier but *G1* a été accompli par la procédure complexe *CP2*, les buts *G2* et *G3* ont été réalisés par la procédure primitive observable *OP1* et la procédure complexe *CP3*. Les deux sous-buts de la procédure *CP2* ont été accomplis par les procédures primitives *PP1* et *PP2*, respectivement. Les deux sous-buts de la procédure *CP3* ont été accomplis par les procédures primitives observables *OP2* et *OP3*, respectivement. Seules les procédures primitives définies comme observables correspondent à des actions de l'interface utilisateur. La figure 2.2.B présente la liste de procédures visibles pour la structure présentée à la figure 2.2.A. Le prochain paragraphe explique comment le traceur de modèle permet

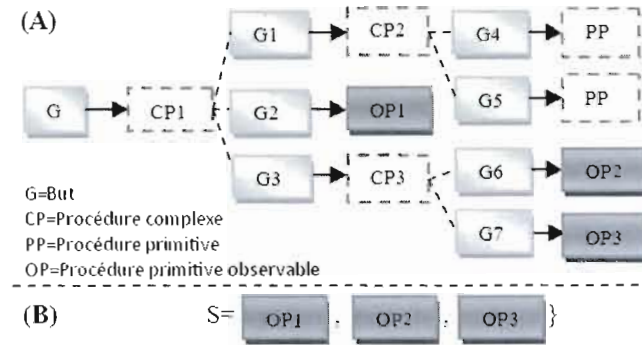


Figure 2.2 Une structure de buts/sous-buts pour un exercice

de suivre le raisonnement d'un apprenant à partir de ses actions percevables.

La tâche principale du traceur de modèle est de trouver une structure de buts/sous-buts telle que celle de la figure 2.2 pour expliquer une séquence d'action d'un apprenant. L'entrée est une liste $S = p1, p2, \dots, pn$ de procédures primitives observables exécutées avec leurs arguments. L'algorithme procède de la façon suivante. L'interpréteur est lancé à partir du but principal de l'exercice. Quand l'interpréteur offre un choix de plusieurs procédures, de buts ou le rappel de plusieurs connaissances sémantiques, l'algorithme sauvegarde l'état de l'interpréteur. Ensuite, une des possibilités est explorée et les autres sont placées dans une pile pour une exploration future. Chaque fois que l'algorithme trouve une possibilité dont la suite d'actions observables s est incluse dans S , mais qu'il ne trouve pas la prochaine action (car l'apprenant n'a pas suivi l'ordre des buts à accomplir prescrit par une procédure complexe) ou si elle est marquée erronée, l'algorithme note la structure actuelle trouvée et la sous-séquence s . Après avoir essayé toutes les possibilités, si une structure pour S n'est pas trouvée, la structure de buts/sous-buts qui respecte la plus grande sous-séquence est retournée (elle est jugée la plus probable, puisqu'elle a le plus grand nombre d'appariement). Pour diminuer la complexité de la recherche, l'algorithme n'explore pas les possibilités qui ne respectent pas une sous-séquence de S , et il ne continue pas plus loin que la dernière étape correcte de l'apprenant.

2.6 L'application du modèle étendu dans CanadarmTutor

Pour modéliser la tâche de manipulation de Canadarm2 avec le modèle étendu, une analyse de tâche a été effectuée. Cette modélisation a été faite sur la base de :

1. des entrevues avec Daniel Dubois, un membre du laboratoire GDAC qui a observé l'entraînement d'astronautes à l'Agence Spatiale Canadienne de Saint-Hubert (Canada) en mars 2006, ainsi que la lecture de son rapport [40] et
2. la lecture des rapports de David Cournoyer, un étudiant ayant assisté à l'entraînement d'astronautes en 2003 [34, 35].

En utilisant ces observations, un premier brouillon des étapes mentales pour déplacer le bras canadien d'une configuration à une autre a été établi. Ensuite, les éléments manquant de l'interface utilisateur de CanadarmTutor ont été notés comme la possibilité de choisir différentes vitesses pour le déplacement du bras canadien. Par la suite, une modélisation a été effectuée avec le modèle cognitif étendu.

En accord avec les recherches sur la cognition spatiale, les connaissances spatiales ont été modélisées par des connaissances sémantiques, sous forme de relations. Pour y arriver, l'espace 3D a été discrétisé en 30 sous-espaces (SE). Il a été conclu que la forme la plus utile de ces zones serait des zones inspirées de configuration du bras. La figure 2.3 présente 6 SE parmi les 16 principaux. Les relations spatiales sont encodées sous forme de concepts composés des types suivants :

1. une caméra peut voir un SE,
2. une caméra donne une vue détaillée/globale d'un module de la SSI,
3. un SE est adjacent à un autre,
4. une caméra est attachée à un module de la SSI.

De plus des concepts primitifs ont été définis pour chaque module, SE, caméra et joint du bras canadien.

Pour certains concepts primitifs importants des connaissances générales ont été définies pour certaines de leur propriétés :

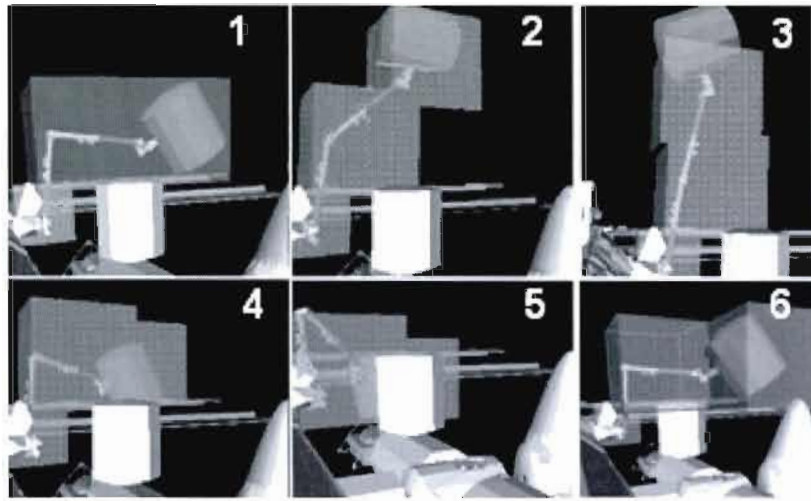


Figure 2.3 Six sous-espaces

1. le nom d'un joint est X et
2. le nom d'un module de la SSI est X.

Les connaissances procédurales pour déplacer le bras d'une position à une autre ont été modélisées comme une boucle où l'apprenant doit effectuer les étapes suivantes :

1. se rappeler des meilleures caméras pour voir les SEs contenant le bras ;
2. sélectionner une caméra donnant une vue globale pour le moniteur 2 et ajuster son pan, tilt et zoom dans cet ordre ;
3. sélectionner une caméra donnant une vue détaillée pour le moniteur 1 et ajuster son pan, tilt et zoom dans cet ordre ;
4. sélectionner une caméra donnant une vue détaillée pour le moniteur 3 et ajuster son pan, tilt et zoom dans cet ordre ;
5. planifier une séquence de SEs pour se rendre du SE courant au but en utilisant le rappel de connaissances générales de SEs adjacents ;
6. déplacer le bras du SE actuel au SE suivant jusqu'à ce qu'un changement de caméra soit nécessaire ou que le but soit atteint.

Les connaissances procédurales manipulant des représentations spatiales peuvent être considérées comme des habiletés spatiales, en accord avec la définition de compétences

spatiales comme étant des habiletés manipulant des représentations spatiales. La modélisation demeure toutefois à un niveau de granularité élevé : il n'a pas été possible de descendre au niveau de détails plus fin de comment choisir un joint et qu'elle valeur de rotation lui appliquer pour déplacer le bras à une position finale. La raison est qu'il n'y a pas de stratégie claire pour choisir les rotations de joints.

Finalement, toutes les connaissances ont été annotées avec des connaissances didactiques telles que des images les illustrants, des indices et des conseils pour leur utilisation dans CanadarmTutor. Ces annotations seront discutées dans la prochaine section sur les services tutoriaux.

2.7 Les services tutoriaux offerts avec le modèle étendu

Cette section présente les principaux services tutoriaux pouvant être offerts grâce au modèle cognitif étendu. Cette section présente ces services dans le contexte de CanadarmTutor. Néanmoins, des services similaires pourraient être implémentés pour d'autres domaines, et des variantes de ces services pourraient être créées.

2.7.1 Exploration libre des connaissances

Un premier service offert à l'apprenant est la possibilité de consulter librement un certain nombre d'informations sur la station spatiale et la tâche de manipulation de Canadarm2, afin de se familiariser avec les connaissances procédurales et sémantiques. Cette fonction a été implémentée sous la forme d'un module nommé « Explorateur de connaissances » accessible dans CanadarmTutor et illustré à la figure 2.4.A. Ce module puise ses informations dans les connaissances décrites pour la modélisation et les ressources didactiques qui les annotent. Dans la section « ISS Modules » (cf. fig. 2.4.B) par exemple, une image illustre chaque module de la SSI. De plus, les relations spatiales encodées permettent d'afficher à l'apprenant les meilleures caméras pour voir le module sélectionné ainsi que les caméra(s) qui y sont attachée(s). Une seconde section nommée « Canadarm joints » illustre les joints de Canadarm2 (cf. fig. 2.4.C). Une troisième section dédiée aux caméras

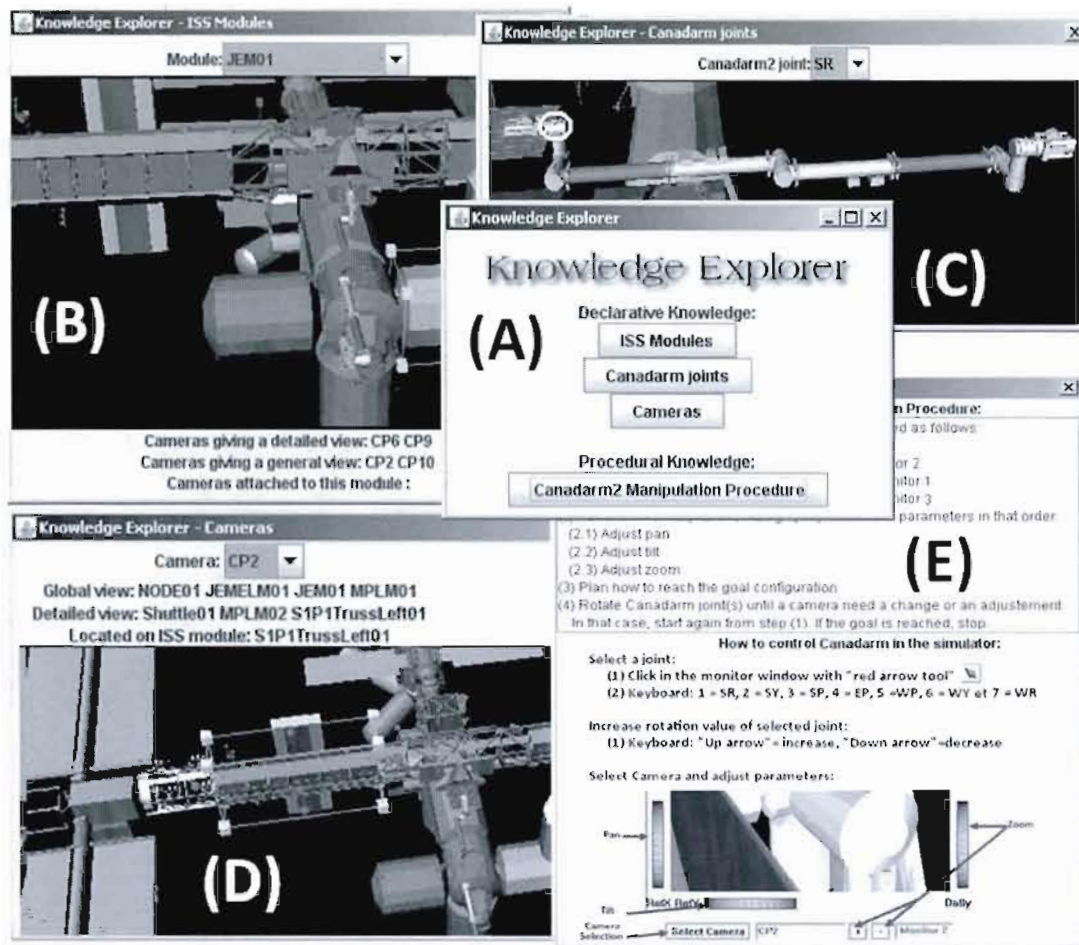


Figure 2.4 Le module « Knowledge Explorer »

(cf. fig. 2.4.D) permet de sélectionner une caméra, de visualiser son module d'attache, et donne une liste des modules dont la caméra donne une vue globale ou détaillée. Encore une fois, ces informations sont celles encodées dans le modèle de connaissances associé à CanadarmTutor. Finalement, une section offre à l'utilisateur une explication des connaissances procédurales pour le déplacement de Canadarm2 (cf. fig. 2.4.E).

2.7.2 Évaluation des connaissances par des exercices procéduraux

Le deuxième service tutoriel est l'évaluation des connaissances par des exercices de résolution de problème. Dans CanadarmTutor ces exercices consistent à déplacer Canadarm2

de sa position initiale à un SE donné, en respectant le protocole de sécurité. Après chaque action d'un apprenant considéré comme l'exécution d'une procédure primitive, le traceur de modèle tente de trouver des structure(s) de buts/sous-buts expliquant la solution partielle de l'apprenant. Ainsi, CanadarmTutor détecte les procédures valides utilisées ainsi que des erreurs procédurales de trois types :

1. l'apprenant a appliqué une procédure erronée pour son but courant,
2. l'apprenant ne respecte pas les contraintes d'ordre entre les sous-buts d'une procédure complexe ou
3. l'apprenant oublie de réaliser un sous-but.

Dans CanadarmTutor, les erreurs procédurales ainsi détectées sont les suivantes :

1. déplacer le bras sans avoir ajusté le moniteur [1, 2 ou 3],
2. ne pas ajuster le moniteur 2 avant le moniteur [1 ou 3],
3. ne pas ajuster le moniteur 1 avant le moniteur 3,
4. ne pas ajuster le pan avant le [tilt ou zoom] pour le moniteur [1, 2 ou 3],
5. ne pas ajuster le tilt avant le zoom pour le moniteur [1, 2 ou 3] et
6. échouer à accomplir un exercice.

Hormis les erreurs procédurales, les connaissances rappelées sont aussi détectées en examinant les structure(s) de buts/sous-buts, puisque ces dernières indiquent si des rappels de connaissances générales ont lieu. Dans CanadarmTutor, deux types de connaissances générales sont ainsi évaluées :

1. les connaissances de type « une caméra peut voir un SE » sont évaluées lorsque l'apprenant choisit une caméra et
2. les connaissances de type « un SE est adjacent à un autre » sont évaluées par les déplacements de l'apprenant d'un SE à un autre.

La détection des erreurs et connaissances valides utilisées, procédurales ou sémantiques, permet de mettre à jour le profil de l'apprenant. Dans le profil de l'apprenant, un indicateur de maîtrise est utilisé pour indiquer la probabilité qu'un apprenant maîtrise chaque

connaissance. Initialement, ces indicateurs sont initialisés avec une valeur par défaut (cette valeur est fixée par défaut à 0.4 dans l'implémentation actuelle, mais le modèle de l'apprenant pourrait être initialisé avec des valeurs représentant un apprenant typique, par exemple). Ensuite, pour chaque utilisation correcte d'une connaissance l'indicateur de maîtrise associé est incrémenté (l'incrément est de 0.05 dans l'implémentation). Dans le cas de l'utilisation d'une connaissance inappropriée, le niveau de maîtrise de cette connaissance et celui des connaissances correctes qui auraient pu être utilisées à sa place sont décrémentés. Ainsi, au fil de la résolution d'exercices, le modèle de l'apprenant est mis à jour pour refléter les forces et les faiblesses de l'apprenant. Il est important de noter que l'implémentation du modèle de l'apprenant est rudimentaire, car il n'est pas l'objet principal de cette recherche. Pour la conception du modèle de l'apprenant, le seul objectif était de construire un modèle minimal pour être capable de tester les autres contributions de la thèse. Dans des travaux futurs, le mécanisme de mise à jour du modèle de l'apprenant pourrait être amélioré pour mieux gérer les probabilités.

La figure 2.5 présente l'outil pour visualiser le modèle de l'apprenant en temps réel. Chaque ligne de la table représente une connaissance. Les quatre premières colonnes indiquent respectivement le nom de la connaissance, le type de la connaissance (ex. : un joint du bras canadien), la valeur de l'indicateur de maîtrise de la connaissance par rapport au niveau requis par la formation (par défaut 0.6) et le nombre de mise à jour du niveau de maîtrise. La cinquième colonne indique pour chaque connaissance sémantique valide, le nom des connaissances erronées qui ont été confondues avec celle-ci. Similairement, pour les connaissances procédurales, cette colonne indique pour chaque procédure valide, les procédures incorrectes qui ont été utilisées à sa place. La sixième colonne n'est pas utilisée par le modèle cognitif et ne sera pas présentée dans ce chapitre.

2.7.3 Évaluation des connaissances générales avec des questions

Le troisième service tutoriel est l'évaluation des connaissances générales avec des questions. Ces questions sont posées à l'utilisateur par séries de cinq lorsque ce dernier clique sur le bouton « Take Knowledge Quiz » de l'interface de CanadarmTutor. Les questions

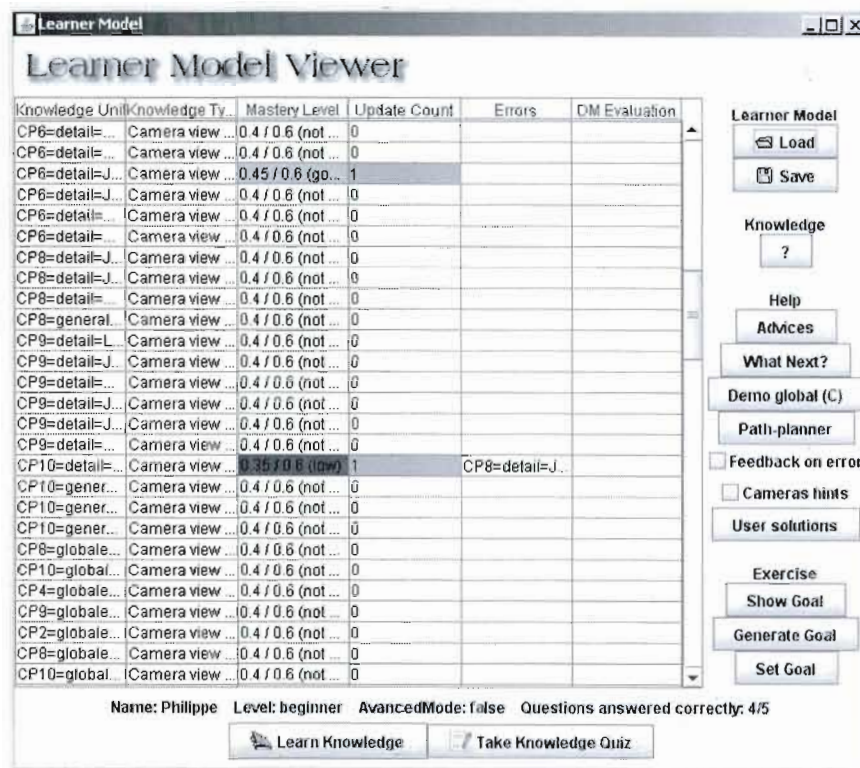


Figure 2.5 Le fenêtre du modèle de l'apprenant

sont générées automatiquement pour tester les connaissances dont l'écart de maîtrise entre le niveau estimé et le niveau requis est élevé. Cinq types de questions sont implémentés pour tester cinq types de connaissances (cf. figure 2.6) :

1. Sur quel module de la SSI est située la caméra X ? (cf. figure 2.6.A)

Une question de ce type teste une connaissance générale de type « une caméra est attaché à un module de la SSI ».

2. Quel est le nom de ce module de la SSI ? (cf. figure 2.6.B)

Une question de ce type teste une connaissance générale de type « le nom d'un module de la SSI est X ».

3. Quelle caméra a été utilisée pour voir ce module de la SSI ? (cf. figure 2.6.C)

Une question de ce type teste une connaissance générale de type « le nom d'un joint est X ».

4. Quel caméra doit être utilisée pour voir cette zone ? (cf. figure 2.6.D)

Une question de ce type teste une connaissance générale de type « une caméra peut voir un SE ».

5. Quel est le nom de ce joint du bras canadien ? (cf. figure 2.6.E)

Une question de ce type teste une connaissance générale de type « le nom d'un joint de Canadarm est X ».

Pour chaque question, l'apprenant doit choisir la bonne réponse parmi un ensemble de choix et peut demander un indice en cliquant sur un bouton. Un indice est fourni sous la forme de la suppression d'un certain nombre de choix erronés. Suite à la réponse de l'apprenant à une question, le niveau de maîtrise de la connaissance testée et des connaissances erronées associées est mis à jour dans le modèle de l'apprenant (tel qu'expliqué dans la sous-section précédente). Il est important de noter que certaines connaissances générales ne sont évaluées que par des questions. Ces questions portent principalement sur les positions relatives des modules et caméras.

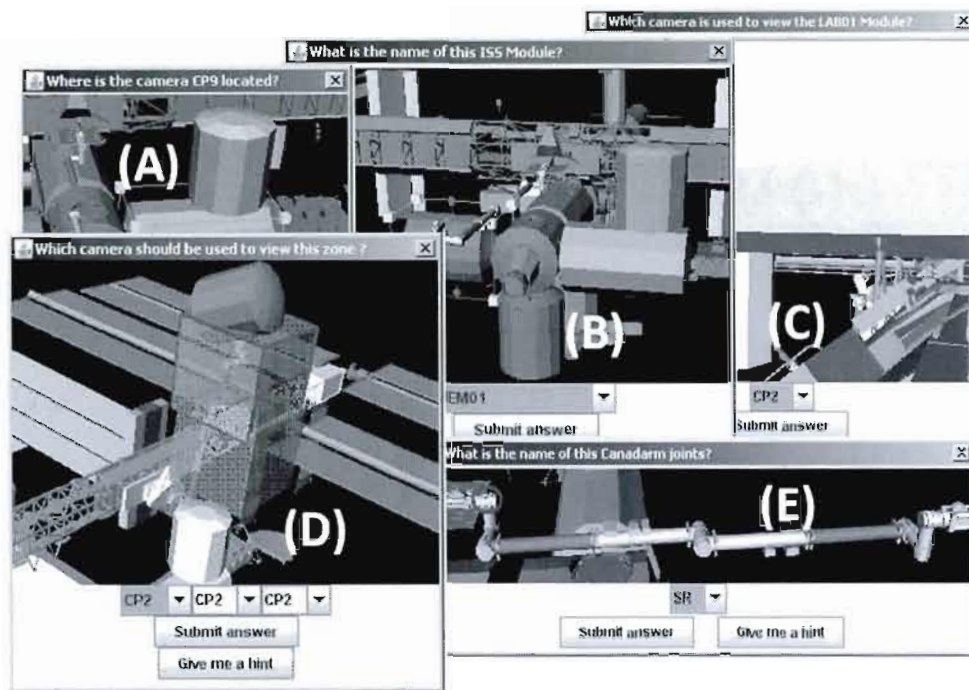


Figure 2.6 Les types de questions du « Space Quiz »

2.7.4 Génération d'indices, démonstrations et conseils personnalisés

Le quatrième service tutoriel est la génération d'indices, de démonstrations et de conseils, à partir des informations contenues dans la modélisation. Dans les activités de résolution, ces messages personnalisés sont offerts à l'apprenant sur demande.

En premier lieu, des indices sur les prochaines actions à effectuer sont présentés à l'utilisateur (cf. fig. 2.7.A) lorsqu'il clique sur le bouton « What next ? » de l'interface de CanadarmTutor. Un indice est généré grâce à l'analyse de la structure de buts/sous-buts générée par le traceur de modèle. Par exemple, si l'utilisateur n'a pas encore déplacé le bras canadien, ni ajusté les moniteurs, il lui sera suggéré de commencer par sélectionner une caméra pour le moniteur 2 et d'ajuster ses paramètres.

Deuxièmement, des démonstrations sont générées automatiquement sur demande. Une démonstration est présentée à l'utilisateur sous forme d'une liste ordonnée de SE à

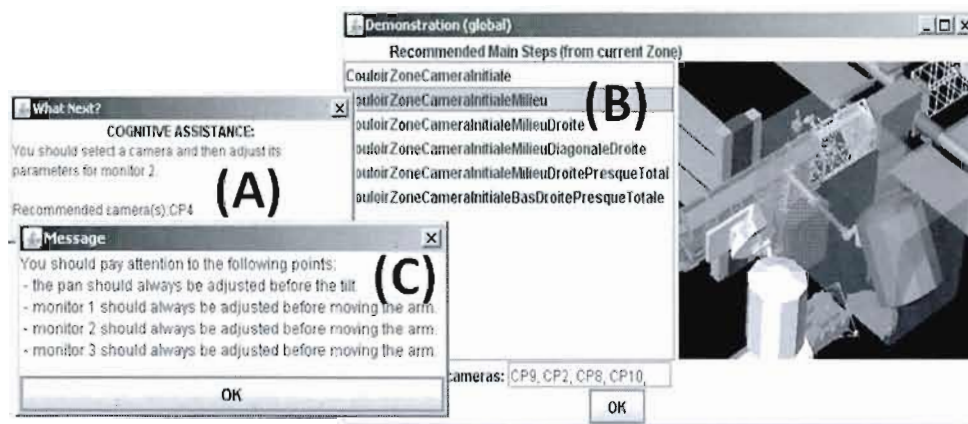


Figure 2.7 La génération de conseils, démonstrations et indices

traverser pour atteindre le SE but à partir du SE contenant le bras canadien (cf. fig. 2.7.B). Une démonstration est générée pour utiliser les chemins les plus courts en terme de nombre de SE traversés et favoriser le passage par des SE avec lesquels l'utilisateur est familier. Ceci est réalisé grâce aux niveaux de maîtrise présents dans le modèle de l'apprenant pour les connaissances générales de type « un SE est adjacent à un autre SE ». De plus, pour chaque SE, la liste des caméras appropriées est affichée grâce aux connaissances générales de type « une caméra peut voir un SE ».

Troisièmement, lorsque l'apprenant clique sur « Advices » dans l'interface de Canadarm-Tutor, un ensemble de conseils personnalisés lui est présenté (cf. fig. 2.7.C). Ces conseils portent sur les connaissances les moins bien maîtrisées dans le modèle de l'apprenant. Les conseils sont des messages textuels extraits des annotations des connaissances de la modélisation.

2.7.5 Génération d'exercices personnalisés

Le cinquième service tutoriel est la génération d'exercices personnalisés. Après la réalisation de plusieurs exercices par l'apprenant, CanadarmTutor acquiert un portrait détaillé des forces et faiblesses de l'apprenant. Le système utilise cette information pour générer des questions et exercices procéduraux en fonction des connaissances à acquérir.

Par exemple, si le système infère qu'un apprenant possède la connaissance fausse que « CP10 » est une bonne caméra pour voir le module JEM, il générera probablement une question sur la connaissance vraie équivalente, ou un exercice demandant son rappel. La génération d'exercices est réalisée grâce à une structure de curriculum qui mentionne l'ensemble des connaissances à maîtriser pour remplir les objectifs de la formation [54].

2.7.6 Rétroaction immédiate et aide proactive

Le sixième et le septième service tutoriel sont respectivement l'offre d'une rétroaction immédiate et l'offre d'une aide proactive pendant les exercices de déplacement de Canadarm2. Ces services peuvent être activés ou désactivés par l'apprenant en cliquant sur "Feedback on error" ou "Camera hints" de l'interface utilisateur. Le premier type d'aide est une rétroaction à l'utilisateur sous forme d'un message après chaque erreur procédurale commise. Ce type de rétroaction est possible grâce au travail de détection d'erreurs procédurales du traceur de modèle expliqué à la section 2.7.2, et est analogue à la rétroaction offerte par les tuteurs cognitifs d'Anderson et al. [10, 13].

Le second type d'aide indique à l'apprenant par un message lorsqu'il devrait changer une ou plusieurs caméras pour obtenir une meilleure vue de la scène d'opération. Ceci est aussi réalisé grâce au traceur de modèle qui permet de savoir quand les connaissances de type « une caméra peut voir un SE » doivent être rappelées (passer à un nouveau ZE peut faire que les caméras doivent être changées).

Ces deux types d'aide sont des exemples d'aide où le système intervient sans avoir été sollicité par l'apprenant (il faut seulement que les services tutoriels soient activés). Il aurait néanmoins été possible de les implémenter autrement. Dans l'implémentation, il a été choisi d'offrir une rétroaction immédiate, car c'est une stratégie tutorielle qui s'est montrée efficace dans les tuteurs cognitifs [10, 13].

2.8 Discussion

Cette section met en contexte l'approche proposée avec les méthodes d'évaluation des connaissances dans les autres STI.

L'évaluation des connaissances sémantiques d'un apprenant est généralement réalisée avec des questions directes (par exemple, [117]). Il existe néanmoins d'autres approches comme évaluer automatiquement une carte de concepts dessiné par un apprenant en la comparant avec la solution d'un expert [150]. Bien que ces approches soient efficaces, elles conviennent mieux à des domaines non procéduraux.

En contrepartie, l'évaluation des connaissances procédurales est généralement accomplie dans le cadre d'exercices de résolution de problèmes. Les systèmes les plus connus de ce type, et basés sur des théories cognitives, sont les Cognitive Tutors [10, 13]. Ces derniers sont basés sur l'hypothèse que la cognition peut être simulée par un système de règles de production manipulant des symboles [12]. Dans les Cognitive Tutors, chaque action d'un apprenant est considérée comme l'application d'une règle effectuant une action ou instanciant des sous-buts. La détection d'erreur se fait en comparant les actions de l'apprenant avec une description de tâche. Bien que ces systèmes obtiennent un grand succès, ils sont centrés sur l'enseignement des connaissances procédurales dans un contexte de résolution de problèmes. En effet, les concepteurs des Cognitive Tutors prennent pour hypothèse que les apprenants ont acquis les connaissances déclaratives ou qu'elles seront disponibles pendant les exercices et que les apprenants sauront quand les utiliser [13].

Contrairement aux deux approches précédentes, la position adoptée dans cette thèse est qu'une évaluation plus juste des connaissances sémantiques et procédurales est réalisée en rendant explicites les connaissances sémantiques qu'un apprenant doit acquérir, et en évaluant ces connaissances avec des questions, mais aussi en contexte de résolution de problèmes. Ceci est en accord avec les experts en éducation qui soutiennent qu'il est important de comprendre comment les connaissances procédurales et sémantiques sont liées dans des tâches procédurales [145]. En ce sens, une critique de l'approche des Cog-

nitive Tutors est qu'elle est limitée, car elle suppose que les connaissances sémantiques peuvent être enseignées efficacement de façon explicite. Or ceci n'est pas toujours le cas. Par exemple, bien qu'on puisse enseigner des connaissances spatiales explicitement (des connaissances sémantiques), manipuler correctement ces représentations dans des tâches procédurales est acquis par l'expérience. Pour certaines tâches, il est donc important pour un STI de pouvoir évaluer l'influence de la maîtrise des connaissances sémantiques sur la performance dans les exercices procéduraux. Ne pas en tenir compte, comme c'est le cas dans les Cognitive Tutors, peut avoir pour effet que des erreurs commises par un apprenant soit interprétées en termes de connaissances procédurales et que cela engendre un feed-back inapproprié.

L'avantage de l'approche proposée ici est que le rappel des connaissances sémantiques est mesuré. Dans le cadre de CanadarmTutor, les connaissances spatiales sont modélisées explicitement et leur rappel dans une tâche procédurale est évalué. Cette représentation explicite des connaissances permet de générer des questions directes sur ces connaissances. Par exemple, pour tester le concept « CP10 donneUneVueDe SE1 » signifiant que la caméra CP10 permet de voir le SE nommé SE1, CanadarmTutor pourrait générer les questions « ? donneUneVueDe SE1 », « CP10 ? SE1 » et « CP10 donneUneVueDe ? ». Si cette connaissance générale n'était pas modélisée explicitement, répondre à ces questions serait vu comme utilisant au moins 3 procédures différentes et le système ne pourrait comprendre qu'elles portent sur la même connaissance sémantique, et que cette connaissance peut être également évaluée en contexte d'exercices procéduraux.

2.9 Conclusion

Ce chapitre a proposé, sur la base des recherches en cognition spatiale, un modèle cognitif pour l'évaluation des habiletés et représentations spatiales dans les STI. Ce modèle est le premier modèle proposé à cette fin. Pour développer le modèle, il a été choisi d'étendre un modèle cognitif pour les STI puisque les modèles de la cognition spatiale demanderaient des efforts considérables pour être adaptés aux contextes des STI.

Le modèle étendu est basée sur l'idée que les connaissances spatiales allocentriques sont des connaissances sémantiques. Il propose d'expliciter les connaissances spatiales pour modéliser leur rappel en contexte procédural. Ceci permet d'évaluer les habiletés et connaissances spatiales directement par des questions ou indirectement en contexte de résolution de problème. Le modèle a été appliqué dans CanadarmTutor, et plusieurs services tutoriels ont été implémentés pour montrer ses possibilités d'exploitation. Une évaluation empirique des services tutoriels offerts dans CanadarmTutor sera présentée au chapitre 6, dans le cadre de l'évaluation du modèle hybride.

Au niveau de la généricité, le modèle étendu présenté dans ce chapitre devrait être réutilisable pour plusieurs autres domaines. La principale raison est qu'il est basé sur Miace, qui lui est basé sur ACT-R, une architecture cognitive, appliquée pour modéliser plus de cent tâches allant de jouer au backgammon à faire des découvertes scientifiques [38]. D'ailleurs, les Cognitive Tutors qui sont également basés sur ACT-R, mais n'offrent pas de mécanisme de rappel, ont été appliqués pour construire des STI pour de nombreux domaines.

Sur l'aspect de l'évaluation du raisonnement spatial, le modèle est plus susceptible d'être réutilisable pour des tâches où le nombre de connaissances spatiales à modéliser n'est pas trop grand. Pour des tâches où le nombre serait trop grand, une solution est d'imposer une structure aux tâches pour limiter la liberté de l'apprenant et ainsi simplifier le travail de modélisation. Structurer une tâche est une stratégie efficace et régulièrement utilisée dans le domaine des STI pour rendre une tâche mieux définie (par exemple, dans EER-Tutor/Kermit [114] tel qu'expliqué à la section 1.3.2). Dans CanadarmTutor, une telle stratégie a été adoptée : il a été entre autres décidé que le bras canadien demeure ancré à sa base mobile et que la position de cette base demeure fixe. Ces restrictions permettent d'offrir des exercices procéduraux tout de même intéressants tout en simplifiant grandement la modélisation. Éventuellement, d'autres exercices avec des restrictions différentes pourrait être ajoutés à CanadarmTutor pour offrir une gamme plus riche d'exercices procéduraux couvrant davantage de situations.

2.9.1 Travaux futurs et extensions

Pour des travaux futurs, un certain nombre d'idées sont envisagées pour améliorer le modèle cognitif, son intégration et son évaluation. En premier lieu, il serait possible d'intégrer à CanadarmTutor de nouveaux types d'exercices et de créer une interface plus raffinée pour les services tutoriels.

En second lieu, il serait intéressant de développer un modèle tenant compte d'autres aspects de la cognition spatiale comme les compétences spatiales et les préférences spatiales de chaque individu. Les compétences spatiales sont des habiletés cognitives que chaque individu possède à différents degrés comme effectuer des rotations mentales d'objets ou imaginer des changements de perspectives. Les compétences spatiales sont généralement évaluées par des tests psychologiques [85, 98]. Considérer les habiletés spatiales des apprenants, ainsi que leurs préférences et compétences spatiales permettrait une personnalisation plus poussée. Il a été montré, par exemple, qu'offrir des instructions allocentriques ou égocentriques à des humains selon leurs préférences peut améliorer leur performance dans des tâches de navigation [133]. De plus, sur la base d'un questionnaire pour évaluer les compétences spatiales, Milik et al. [113] ont observé qu'il semble avantageux d'adapter la présentation visuelle d'activités d'apprentissage avec du texte ou des images dans le contexte d'exercices de raisonnement spatial dans un environnement à deux dimensions. Des expériences préliminaires par Morganti et al. [118] montrent aussi qu'évaluer la compétence d'orientation spatiale chez des patients avec des dommages cérébraux dans des environnements virtuels donne des résultats similaires à ceux obtenus avec des tests psychologiques sur papier. Cela pourrait donc être une piste intéressante à explorer.

CHAPITRE III

LES APPROCHES D'ACQUISITION AUTOMATIQUE DE CONNAISSANCES DU DOMAINE

Le modèle cognitif proposé au chapitre précédent permet d'offrir plusieurs services tutoriels, mais il est conçu pour être appliqué pour des domaines bien définis, modélisables avec des règles de production. Dans le cadre de CanadarmTutor, par exemple, le modèle permet de représenter en partie le raisonnement spatial crucial au déplacement de Canadarm2. Néanmoins, il ne décrit pas la façon de choisir les rotation de joints à effectuer. La raison est qu'il n'y a pas de stratégie claire et facilement formalisable pour déterminer les rotations de joints à effectuer pour atteindre une configuration donnée (il n'y en a pas d'enseigner à l'Agence Spatiale Canadienne [40]). Aussi, une limite de l'utilisation de modèles cognitifs et de l'approche du traçage de modèle en général est que créer un modèle de tâche à la main est coûteux en temps.

Pour pallier ces limites et celles des approches classiques présentées au chapitre 1, ce chapitre et les deux suivants s'intéressent à la conception d'algorithmes pour l'acquisition automatique de connaissances du domaine. Tout d'abord, une revue de littérature des approches d'acquisition automatique des connaissances dans les STI, en intelligence artificielle et en modélisation cognitive en général est présentée. Pour chaque approche mentionnée, les limites et principales caractéristiques sont identifiées. Puis, à la fin du chapitre, un ensemble d'objectifs pour la conception d'une nouvelle approche est énoncé.

Le chapitre 4 propose ensuite une nouvelle approche pour l'acquisition automatique de

connaissances du domaine dans les STI, et le chapitre 5, en présente une extension. Puis, le dernier chapitre de la thèse, le chapitre 6, présente un modèle hybride composé de cette nouvelle approche et du modèle cognitif décrit au chapitre 2, pour tirer profit du meilleur des deux approches.

3.1 Les travaux sur l'acquisition automatique des connaissances en intelligence artificielle et en modélisation cognitive

La construction d'agents intelligents ou de systèmes d'intelligence artificielle capables d'apprendre par l'observation d'autres agents (virtuels ou humains) est un problème qui a été étudié par plusieurs chercheurs en intelligence artificielle. Néanmoins, les solutions proposées reposent généralement sur des hypothèses fortes. Cette section en présente deux exemples représentatifs.

3.1.1 Le travail de vanLent et Laird

van Lent et Laird [157] ont proposé un mécanisme pour l'apprentissage de règles de production à partir de comportements enregistrés par des humains. Selon cette approche, un humain enseigne à un agent par la démonstration de l'accomplissement d'une tâche. Toutefois, cette approche est liée à une conception très spécifique de l'intelligence, car les démonstrateurs doivent spécifier leurs actions en termes d'opérateurs complexes organisés en hiérarchie et ayant des conditions de satisfaction, et les humains doivent indiquer leurs intentions tout au long d'une démonstration. Cette approche est donc contraignante sur la façon d'encoder les démonstrations et suppose que les tâches apprises peuvent être représentées par des ensembles de règles de production.

3.1.2 Le travail de Langley et Ohlsson

Une seconde approche pour l'extraction de modèles cognitifs pour des tâches est celle de Langley et Ohlsson [100]. Cette approche est basée sur l'hypothèse que la résolution d'un problème est une recherche dans un espace d'états [122] et que tout comportement

cognitif peut être représenté comme l'application de règles de production [123]. À partir de la définition d'un espace d'états contenant un état initial et final et un ensemble d'opérateurs de transition, l'approche proposée permet de découvrir un ensemble de règles de production indiquant comment choisir les opérateurs pour traverser l'espace d'états et atteindre un état final. L'algorithme utilise chaque cheminement donné pour raffiner les règles de façon à mieux les classer par rapport aux autres chemins de l'espace d'états. L'approche a été appliquée avec succès pour des domaines simples comme la soustraction multi-colonnes. Comme pour l'approche précédente, cette dernière demande la spécification des opérateurs dans un format propre, et se commet à une conception particulière de la cognition. L'approche est aussi coûteuse, car son application demande de parcourir entièrement un espace d'états.

3.2 L'apprentissage de connaissances du domaine dans les STI

Les travaux cités dans la section précédente sont un aperçu des approches d'acquisition automatique de connaissances proposées en intelligence artificielle et en modélisation cognitive. En plus des limites mentionnées, pour le contexte de cette thèse, ils ont l'inconvénient de ne pas avoir été conçus pour les STI. Dans le domaine des STI, plusieurs chercheurs ont proposé des approches pour l'apprentissage automatique de modèles de tâche/espaces problèmes [18, 20, 91, 108, 110, 138, 148, 149] ou de contraintes [138, 148, 149].

3.2.1 Le système Demonstr8 de Blessing

Un premier travail est le système Demonstr8 de Blessing [20]. Il permet l'apprentissage de règles de production pour la construction de tuteurs cognitifs à partir de démonstrations effectuées par des humains. Une règle de production est inférée sur la base de paires entrée(s) et sortie(s) en utilisant le mécanisme d'analogie de l'architecture cognitive ACT-R [12]. Après la découverte de règles de production, un auteur peut les annoter avec des habiletés et des indices. Une première limitation de Demonstr8 est que les règles de production inférées sont des règles simples à deux arguments et que leur

partie action est constituée d'au plus une fonction élémentaire LISP prédéfinie comme multiplier ou soustraire. Le succès de l'application de l'approche dépend donc du fait qu'une tâche peut-être modélisée par des règles et qu'un auteur fournira les fonctions LISP appropriées pour les constituer. De plus, lors des démonstrations, un humain doit spécifier ses buts et sous-buts pour permettre à Demonstr8 de lier les règles générées, ce qui ajoute un travail aux démonstrateurs. Pour valider son approche, Blessing [20] a appliqué Demonstr8 au domaine simple de la soustraction multicolonne et a admis que pour des domaines plus complexes où des raisonnements complexes et cruciaux ne sont pas effectués dans l'interface graphique, comme c'est le cas dans CanadarmTutor, l'approche serait difficilement applicable.

3.2.2 Le Behavior Recorder de McLaren et al.

Un autre travail est le Behavior Recorder [11, 110], un outil intégré au logiciel CTAT [10]. En mode démonstration, le Behavior Recorder permet à un auteur de définir un problème et d'en démontrer les différents cheminements corrects et erronés. Le Behavior Recorder construit alors progressivement un espace d'états au cours des démonstrations, où les états représentent des états du problème et les transitions représentent les actions pour passer d'un état à un autre. Un auteur peut annoter les transitions avec des indices et des messages. Ensuite, CTAT peut utiliser l'espace problème créé pour présenter des indices et messages aux apprenants qui effectueront le problème. Une limite considérable du Behavior Recorder est qu'il n'y a pas de généralisation. En effet, le Behavior Recorder intègre les solutions pour chaque problème directement dans un espace problème qui lui est spécifique.

3.2.3 Le travail de Jarvis et Nuzzo-Jones

Jarvis et Nuzzo-Jones [91] offrent une alternative au Behavior Recorder de CTAT. Ils ont conçu un système pour extraire un ensemble de règles de production générales à partir d'un ensemble de démonstrations, étant donné un ensemble minimal de connaissances sur le domaine. Cet ensemble de connaissances consiste en des conditions et des fonctions

pour la création respectivement de la partie condition et action des règles. Pendant une démonstration, le démonstrateur doit annoter les valeurs qu'il manipule dans l'interface graphique avec une sémantique (par exemple, indiquer qu'un nombre est une retenue dans le contexte d'une soustraction multicolonne). Les principales limites de l'approche de Jarvis et Nuzzo-Jones sont que (1) la recherche de règles est une recherche exhaustive dont la complexité croît rapidement avec le nombre de fonctions et de conditions à considérer, (2) les règles obtenues ont tendance à être trop générales, et (3) l'approche a été appliquée seulement à des domaines simples et bien définis comme l'addition de fractions et le jeu de Tic-Tac-toe [91].

3.2.4 SimStudent de Matsuda et al.

Un autre travail est l'agent virtuel SimStudent [108]. Ce dernier est capable d'apprendre des règles de production en observant un humain résoudre des problèmes. De façon similaire au travail de Jarvis et Nuzzo-Jones, SimStudent requiert de définir préalablement un ensemble de connaissances du domaine. Ces dernières sont ici (1) des prédicats pour composer la partie condition des règles et (2) des opérateurs pour former la partie action des règles. Pendant une démonstration, un démonstrateur doit spécifier le nom de chaque règle dont il fait la démonstration et l'objet de son attention sur l'écran (les éléments manipulés par chaque règle). Pour chaque démonstration d'une règle, SimStudent détermine les prédicats caractérisant ce qui est visible à l'écran, puis tente de trouver la combinaison d'opérateurs la plus courte pour générer l'action démontrée, et s'assure que cela est cohérent avec toutes les démonstrations antérieures de la règle. Une fois des règles apprises, SimStudent peut les appliquer sur de nouveaux problèmes pour en trouver la solution. SimStudent a été appliqué dans le domaine de l'algèbre. Une critique de SimStudent est qu'il suppose que le processus de résolution peut-être modélisé avec un nombre restreint de règles, ce qui n'est pas nécessairement évident pour des domaines mal définis, et que fournir les connaissances préalables du domaine peut être une tâche difficile.

3.2.5 Le travail de Barnes et Stamper

Un autre travail du domaine des STI est celui de Barnes et Stamper [18], appliqué pour le domaine des preuves en logique. Leur approche consiste à construire un processus de décision de Markov contenant les solutions d'apprenants à un problème. Cela est principalement un graphe où chaque nœud représente un état correct ou erroné dans la résolution d'un problème et chaque lien une action pour passer d'un état à un autre. Ensuite, à partir d'un état, un chemin optimal peut être calculé pour atteindre un but, afin d'offrir des indices à un apprenant. L'optimalité peut-être la fréquence, la probabilité de ne pas commettre une erreur et de minimiser le nombre d'actions. Cette approche ne demande pas de spécifier des connaissances du domaine sinon que de définir les actions à enregistrer. Par contre, puisque tous les chemins sont stockés dans un graphe, cette approche ne semble applicable que pour des domaines où le nombre de possibilités est limité. De plus, il n'est pas toujours bénéfique d'un point de vue pédagogique de suggérer la solution optimale à un apprenant ; il est plutôt souhaitable d'offrir des indices personnalisés pour faire progresser un apprenant dans le continuum de novice à expert. L'approche de Barnes et Stamper [18], comme les précédentes, ne tient pas compte des profils des apprenants.

3.2.6 CAS de Suraweera et al.

Une approche qui se distingue de celles présentées jusqu'ici est CAS de Suraweera [148, 149] qui vise l'apprentissage de contraintes pour créer des STI à base de contraintes. La première phase de cette approche est la création d'une ontologie du domaine. Ensuite, des contraintes syntaxiques sont extraites directement de l'ontologie (à partir par exemple des restrictions sur les attributs, présentes dans l'ontologie). Par la suite, un auteur doit créer des problèmes et des solutions annotées avec des instances de concepts de l'ontologie. Ceci permet à CAS de générer des contraintes sémantiques. CAS a obtenu un certain succès pour des domaines comme l'addition de fractions et la modélisation de diagrammes entité-relation. Néanmoins, l'applicabilité et le succès de cette approche

dépend directement de la possibilité de construire une ontologie appropriée et suffisamment détaillée pour un domaine. Or, pour de nombreux domaines cela peut-être très difficile, voire même impossible. De plus, construire une ontologie demande une certaine expertise [51].

3.2.7 Le travail de Riccuci et al.

Une approche similaire à CAS est le travail de Riccuci et al. [138], qui utilise un mécanisme de fouille de données basé sur un sous-ensemble de Prolog. Avec l'approche de Riccuci et al., un auteur doit fournir en Prolog (1) une ontologie d'un domaine, (2) un ensemble d'exemples positifs et négatifs ainsi que (3) un ensemble de connaissances préalables du domaine. Le cadre d'application permet de découvrir un ensemble de contraintes sous forme de règle « SI-ALORS » devant être ensuite validée manuellement. Riccuci et al. [138] démontrent que leur approche a généré des contraintes plausibles pour l'évaluation de solutions exprimées sous forme d'équations algébriques de premier ordre sans simplification pour un STI fictif. Toutefois, ce domaine est simple et bien défini, et il n'est pas démontré que l'approche de Riccuci et al. s'applique bien à des domaines plus complexes ou difficile à formaliser. En fait, leur approche n'est applicable qu'à des domaines formalisables en Prolog. De plus, que les contraintes soient exprimées en fonction des entités de l'ontologie limite ce que les contraintes peuvent exprimer. Une autre critique est qu'un auteur doit fournir beaucoup de connaissances préalablement pour appliquer l'approche. Finalement, Riccuci et al. suggèrent que leur approche peut être utilisée pour générer des tuteurs cognitifs, mais ils n'en font pas la démonstration.

3.3 Résumé comparatif des approches présentées et objectifs

Le tableau 3.1 présente une vue récapitulative des caractéristiques des approches décrites dans ce chapitre. En somme, les systèmes actuels possèdent une ou plusieurs des limitations suivantes :

1. une obligation de fournir une quantité importante de connaissances du domaine

- préalablement aux démonstrations : Blessing [20], Jarvis [91], Matsuda et al. [108], Riccuci [138], Suraweera [148, 149], van Lent et Laird [157],
2. une efficacité démontrée que pour des domaines bien définis : Barnes et Stamper [18], Blessing [20], Jarvis [91], Langley et Ohlsson [100], Matsuda et al. [108], McLaren et al. [110], Riccuci [138], Suraweera [148, 149], van Lent et Laird [157] (toutes les approches),
 3. être basé sur l'hypothèse forte qu'une tâche peut être modélisée avec des règles de production : Jarvis [91], Langley et Ohlsson [100], Matsuda et al. [108], Suraweera [148, 149], van Lent et Laird [157],
 4. ne pas tenir compte du profil des personnes effectuant les démonstrations : Barnes et Stamper [18], Blessing [20], Jarvis [91], Langley et Ohlsson [100], Matsuda et al. [108], McLaren et al. [110], Riccuci [138], Suraweera [148, 149], van Lent et Laird [157] (toutes les approches),
 5. apprentissage de connaissances spécifiques à un problème : Barnes et Stamper [18], Jarvis [91],
 6. et obligation d'annoter les démonstrations avec des informations supplémentaires (par exemple, des intentions) : Jarvis [91], Matsuda et al. [108], Suraweera [148, 149].

La plupart de ces limites sont des limites importantes. Pour concevoir une nouvelle approche d'acquisition automatique des connaissances du domaine pour les domaines mal définis, cette thèse se fixe les objectifs suivants :

1. que l'approche ne requiert aucune spécification préalable des connaissances du domaine,
2. qu'elle puisse tenir compte du profil de chaque apprenant ou expert effectuant une démonstration ou utilisant le système,
3. qu'elle puisse enrichir sa base de connaissances après chaque résolution de problème par un apprenant,

Proposition	Données en entrée	Connaissances produites/extraites	Type d'apprentissage	Conception de l'intelligence
Van Lent & Laird [155]	1) Des démonstrations : séquences d'applications d'opérateur annotées avec des intentions. 2) Hiérarchie d'opérateurs avec conditions de satisfaction	1) Un ensemble de règles de production	Procédural, supervisé	Système de règles de production
Langley & Ohlsson [99]	1) Un espace d'états 2) Des cheminements dans l'espace d'états	1) Un ensemble de règles de production qui représentent des heuristiques de recherche dans l'espace d'états	Procédural (stratégies de recherche), supervisé	Règles de production qui contrôlent la recherche dans un espace d'états
Matsuda et al. [107]	1) Des opérateurs et prédicats 2) Démonstrations de règles incluant le nom de la règle démontrée et le centre d'attention	1) Des règles de production « où, quand, comment » générales	Procédural	Système de règles de production
Barnes et Stamper [17]	1) Des séquences d'actions	1) un processus de décision de Markov spécifique à un problème	Procédural	Exploration d'un espace d'états
McLaren et al. [10, 109]	1) Démonstration de solutions pour un problème + annotations	1) un graphe propre au problème qui intègre les solutions démontrées.	Procédural	Exploration d'un espace d'états
Jarvis & Nuzzo-Jones [90]	1) Des fonctions et conditions 2) Des démonstrations de résolution dont les éléments déclaratifs sont annotés avec une sémantique	1) des règles de production générales	Procédural	Système de règles de production
Surawee et al. [146][147]	1) Une ontologie du domaine 2) Un ensemble de problèmes et solutions annotés avec l'ontologie	1) des contraintes syntaxiques et des contraintes sémantiques générales	Sémantique (des contraintes)	
Riccuci et al. [136]	1) une ontologie du domaine 2) des exemples positifs et négatifs 3) des connaissances préalables du domaine	1) des contraintes sous formes de règles exprimées en Prolog	Sémantique (des contraintes)	
Blessing [19]	1) Des fonctions élémentaires 2) des exemples annotés avec les intentions	1) des règles de production générales simples	Procédural	Système de règles de production (ACT-R)

Tableau 3.1 Tableau récapitulatif des approches présentées

4. qu'elle soit applicable pour des domaines procéduraux ayant un large nombre de possibilités et aucune stratégie claire pour résoudre un problème (comme la manipulation de Canadarm2),
5. qu'elle ne suppose aucune vision restrictive de l'intelligence (comme des règles de production)
6. et qu'elle ne demande pas aux démonstrateurs de fournir des informations supplémentaires comme leurs intentions lors des démonstrations.

Les deux prochains chapitres présentent une nouvelle approche qui répond à ces objectifs.

CHAPITRE IV

UNE APPROCHE POUR L'ACQUISITION AUTOMATIQUE DES CONNAISSANCES DU DOMAINE

Pour développer une approche d'acquisition automatique de connaissances du domaine répondant aux objectifs énoncés à la fin du chapitre précédent, il a été choisi d'utiliser des algorithmes de fouille de données. La fouille de données est le domaine qui s'intéresse à la découverte de motifs dans des grands ensembles de données [80]. Ces motifs peuvent être de différentes natures. Dans le contexte de cette thèse, l'objectif est d'appliquer des algorithmes de fouille de données sur un ensemble de traces de résolution de problème pour extraire automatiquement des modèles de tâche afin de supporter des services tutoriels dans un STI. Pour réaliser cet objectif, il faut néanmoins avoir des algorithmes adaptés pour tenir compte des facteurs spécifiques dans lequel un STI comme CanadarmTutor opère. Ces facteurs incluent l'ordre temporel des actions effectuées pour accomplir une tâche, des actions ayant des paramètres et le profil des personnes enregistrant les traces. L'ensemble de ces facteurs suggère qu'une approche d'extraction de motifs temporels est nécessaire.

Selon Han et Kamber [80], il y a quatre principaux types de motifs qui peuvent être extraits de séquences temporelles : les tendances, les séquences similaires, les motifs séquentiels et les motifs périodiques. Les tendances sont des orientations à long terme dans des séquences temporelles. La recherche de séquences similaires inclut principalement deux problèmes : (1) découvrir les séquences parmi un ensemble de séquences temporelles qui contiennent des sous-séquences similaires à une séquence donnée et (2) découvrir dans

un ensemble de séquences temporelles un ou plusieurs groupes de séquences similaires. Les motifs séquentiels sont des sous-séquences apparaissant dans plusieurs séquences temporelles. Finalement, les motifs périodiques sont des motifs apparaissant périodiquement dans des séquences temporelles. Dans ce travail, il a été choisi d'extraire des motifs séquentiels [4] pour découvrir des séquences d'actions apparaissant dans plusieurs traces d'utilisation pour une tâche, et former un modèle de tâche partiel. L'hypothèse sous-tendant ce choix est que les séquences d'actions communes à plusieurs utilisateurs réussissant un exercice pourraient être utilisées pour aider d'autres utilisateurs et que cela favoriserait leur apprentissage de la tâche.

Pour la recherche de motifs séquentiels, plusieurs algorithmes efficaces ont été proposés [75, 81, 166, 144, 169]. Ils ont été appliqués entre autres pour l'analyse de données provenant de sismographes [87] et l'analyse de code source en génie logiciel [167]. Alors que les algorithmes traditionnels de recherche de motifs séquentiels ont pour seul but de découvrir des séquences d'événements apparaissant dans plusieurs séquences d'une base de séquences [4], plusieurs algorithmes offrent de nombreuses extensions pour la recherche de motifs séquentiels tel que :

1. la recherche de motifs séquentiels respectant des contraintes temporelles [87],
2. l'extraction des représentations compactes de motifs séquentiels [103, 143, 167, 162],
3. la mise à jour incrémentale des motifs séquentiels pour une base de séquences qui évolue [167],
4. la recherche de motifs séquentiels respectant des contraintes sous forme d'expressions régulières [74],
5. la recherche de motifs séquentiels dans des bases de données multidimensionnelles [136, 143] ou dans des flots de données (*stream*) [29].

Pour ce travail, un algorithme de recherche de motifs séquentiels fréquents a été adapté pour répondre aux exigences particulières de l'extraction de modèles de tâches dans les

STI [64, 65, 66]. L'algorithme combine les propriétés de plusieurs algorithmes comme accepter des contraintes temporelles [87], traiter des bases de séquences multidimensionnelles [136], extraire une représentation compacte et sans perte d'information de tous les motifs séquentiels [162, 143], et ajoute aussi des propriétés originales comme accepter des symboles avec des valeurs numériques et faire le regroupement des valeurs similaires.

Le reste de ce chapitre est organisé de la façon suivante. La section 4.1 présente le problème de la recherche de motifs séquentiels dans des traces d'utilisation. Ensuite, les sections 4.2 à 4.7 décrivent six limites rencontrées avec les algorithmes classiques, puis les extensions apportées à un de ces algorithmes et d'autres solutions pour les adresser. Les dernières sections présentent (1) les services tutoriels offerts dans CanadarmTutor sur la base des connaissances extraites par cet algorithme, (2) une évaluation des performances dans CanadarmTutor, (3) une revue des autres applications de la recherche de motifs séquentiels dans le domaine des STI et de l'apprentissage en ligne, (4) une autre application de l'algorithme proposé et (5) une conclusion.

4.1 La recherche de motifs séquentiels dans des traces d'utilisation

Pour définir le problème de la recherche de motifs séquentiels dans des traces d'utilisation, il faut tout d'abord décrire le problème de la recherche de motifs séquentiels fréquents en fouille de donnée, défini par Agrawal et Srikant [4].

4.1.1 Le problème de la recherche de motifs séquentiels fréquents

Soit D une base de séquences et I un ensemble d'*items*. Chaque *séquence* s de D est définie comme une suite ordonnée d'*itemsets* (des ensembles d'items) $A_1, A_2, \dots, A_n \subseteq I$. Les items d'un même itemset sont considérés non ordonnés ou simultanés. La figure 4.1.a illustre une base de six séquences. La séquence $S1$ indique par exemple que l'item a a précédé l'item b , qui a été suivi par les items c et d en temps simultané, puis finalement par l'item e .

Une séquence $sa = (A_1, A_2, \dots, A_n)$ est dite contenue dans une autre séquence $sb =$

(B_1, B_2, \dots, B_m) si et seulement s'il existe des entiers $1 \leq i_1 < i_2 < \dots < i_n \leq m$ tel que $A_1 \subseteq B_{i_1}, A_2 \subseteq B_{i_2}, \dots, A_n \subseteq B_{i_n}$. La *fréquence relative* ou *fréquence* d'une séquence sa est définie comme le pourcentage de séquences $s \subseteq D$ tel que $sa \subseteq s$, et est notée $supD(sa)$ ou simplement $sup(sa)$ lorsque le contexte est clair.

Le *problème de la recherche des motifs séquentiels fréquents* est de trouver toutes les séquences sa tel que $supD(sa) \geq minsup$ pour une base de séquences D et un seuil $minsup$ donné. La figure 4.1.b illustre une partie des motifs séquentiels découverts pour $minsup = 33\%$ à partir de la base de séquences de la figure 4.1.a. Par exemple, le motif $M4$ est considéré fréquent, car sa fréquence est de 50% (il est contenu dans les séquences $S1, S2$ et $S4$, soit trois séquences sur six).

4.1.2 La recherche de motifs séquentiels dans des traces d'utilisation

Le *problème de la recherche de motifs séquentiels dans des traces d'utilisation* a été défini dans une recherche antérieure à laquelle l'auteur de ce document a participé [128, 127, 129]. Cette recherche a défini ce problème comme la recherche de motifs séquentiels dans une base de séquences, où chaque séquence est une tentative de résolution d'un exercice procédural, et chaque item représente une action (note : ce sont des séquences d'itemsets tel que définies dans la sous-section précédente : les actions peuvent être groupées en itemset dans les séquences pour signifier que leur ordonnancement n'est pas important). Chaque séquence d'une base de séquences peut être le résultat d'utilisateurs différents. L'objectif de la recherche de motifs séquentiels dans des traces d'utilisation est de découvrir des séquences d'actions effectuées dans plusieurs traces de résolution de problème afin de pouvoir ensuite utiliser ces motifs pour supporter des services tutoriels comme la suggestion des prochaines actions à faire à un apprenant pour résoudre un problème.

Dans cette recherche antérieure, un essai préliminaire a été fait avec CanadarmTutor. 155 actions pouvant être effectuées par un apprenant dans un exercice de déplacement du bras robotisé ont été recensées : (1) sélectionner une caméra, (2) effectuer une pe-

tite, moyenne ou grande rotation positive ou négative d'un joint et (3) faire une petite, moyenne ou grande incrémentation ou décrémentation du pan, tilt ou zoom d'une caméra. Ensuite, une cinquantaine de traces de résolution de problème ont été enregistrées sous forme de séquences dans une base de séquences. Les quelques traces correspondant à des échecs ont été retirées. Pour cette première expérience, l'algorithme de recherche de motifs séquentiels PrefixSpan [134] a été appliqué. PrefixSpan a été choisi, car il peut être facilement étendu. PrefixSpan est aussi un des algorithmes parmi les plus efficaces pour la recherche de motifs séquentiels. PrefixSpan extrait des motifs fréquents en projetant récursivement une base de séquences en bases de séquences plus petites. Chaque motif est construit un item à la fois dans chaque base de séquences projetée en explorant seulement les actions unitaires localement fréquentes.

L'application de PrefixSpan sur les traces enregistrées dans CanadarmTutor a résulté en une centaine de motifs fréquents. Un exemple de motif qui a été découvert est de sélectionner la caméra CP2, qui donne une vue rapprochée du bras canadien dans sa position initiale, de décrétement légèrement le yaw de la caméra CP2, de sélectionner le joint EP du bras canadien, et de décrétement un peu sa valeur de rotation. Ensuite, les motifs ont été intégrés à CanadarmTutor pour suggérer aux apprenants les prochaines actions à faire. Par exemple, si un apprenant ajuste la caméra CP2 et sélectionne le joint EP, CanadarmTutor peut lui suggérer de décrétement la valeur de rotation du joint EP, en vertu du motif précédent. La justification de cette suggestion est que plusieurs apprenants ont réussi l'exercice en effectuant ce motif.

Bien que l'ensemble des motifs extraits avec PrefixSpan constitue un espace problème qui capture différentes façons de résoudre des problèmes, plusieurs limites ont été notées lors de cette première expérience. Les prochaines sections décrivent ces limites et des extensions apportées à PrefixSpan pour les pallier. Les services tutoriels pouvant être offerts seront décrits en détail à la fin de ce chapitre.

ID	Séquence		ID	Motif	Fréquence
S1	a, b, {c d}, e	→	M1	a	83 %
S2	a, d, e		M2	a, e	66 %
S3	e, f, g		M3	b, c	33 %
S4	b, a, d, e		M4	a, d, e	50 %
S5	{a, b}, {c d}, f, g		M5	a, {c, d}	33 %
S6	a, e		...		

Figure 4.1 (a) une base de 6 séquences et (b) des motifs séquentiels extraits

4.2 Extension de la recherche de motifs séquentiels avec des contraintes temporelles

Un premier problème rencontré est que les motifs extraits contiennent souvent des « trous » par rapport aux séquences qui les contiennent. Un trou dans un motif est un ou plusieurs items qui peuvent être ajoutés entre deux items d'un motif tel que le motif résultant est présent dans au moins une des séquences de la base de séquences où le motif a été extrait. Par exemple, dans le motif $M2 = \{a, e\}$ de la figure 4.1, il y a un trou entre les items a et e qui est l'item d , puisque l'item d apparaît dans au moins une séquence entre les items a et e (il apparaît en fait dans trois séquences : $S1$, $S2$ et $S4$).

Un trou de quelques items est acceptable en contexte tutoriel parce que cela élimine un certain bruit (des actions non fréquentes), mais lorsqu'un trou est trop grand, les motifs deviennent difficilement utilisables pour des services tutoriels comme recommander la prochaine action à effectuer à un apprenant. Il est donc essentiel de limiter la taille des trous dans les motifs séquentiels trouvés.

Un second problème est que certains motifs sont trop courts pour être utiles dans un contexte tutoriel (par exemple, les motifs de taille 1). Il devrait donc y avoir une façon de spécifier une longueur minimale pour les motifs à découvrir. Une extension de la recherche de motifs séquentiels pour pallier ces limites est d'extraire des motifs séquentiels respectant des contraintes temporelles et des contraintes sur leur longueur [87].

4.2.1 La recherche de motifs séquentiels avec des contraintes temporelles

Une *base de séquences temporelles* D est définie comme un ensemble de *séquences temporelles* $s = \langle (t_1, X_1), (t_2, X_2), \dots, (t_n, X_n) \rangle$ où chaque itemset X_x est annoté avec une estampille temporelle t_x . Chaque estampille représente le temps écoulé depuis le premier itemset de la séquence. Un exemple de séquence temporelle est $\langle (0, a), (1, bc), (2, d) \rangle$ qui indique que l'item d est apparu une unité de temps après les items b et c , et deux unités de temps après l'item a . L'*intervalle de temps entre deux itemsets* (t_x, X_x) et (t_y, X_y) d'une séquence est défini comme $|t_x - t_y|$. Par exemple, pour la séquence $\langle (0, a), (1, bc), (2, d) \rangle$, l'intervalle de temps entre le premier itemset et le troisième itemset est de deux unités de temps et l'intervalle entre les itemsets $(0, a)$ et $(1, bc)$ est d'une unité de temps. Dans ce travail, pour l'application dans CanadarmTutor, un intervalle d'une unité de temps est supposé entre tous les itemsets adjacents (t_x, X_x) et (t_y, X_y) d'une séquence d'une base de séquences, de façon à ce que l'intervalle de temps entre deux itemsets soit une mesure du nombre d'itemsets suivant l'occurrence du premier, plutôt qu'une mesure de temps écoulé. Néanmoins, tous les exemples présentés dans ce chapitre sont décrits pour le cas général où l'intervalle est non constant.

Une séquence temporelle $sa = \langle (ta_1, Xa_1), (ta_2, Xa_2), \dots, (ta_n, Xa_n) \rangle$ est dite incluse dans une autre séquence temporelle $sb = \langle (tb_1, Xb_1), (tb_2, Xb_2), \dots, (tb_m, Xb_m) \rangle$ si et seulement s'il existe des entiers $1 \leq i_1 < i_2 < \dots < i_n \leq m$ tel que $Xa_1 \subseteq Xb_{i_1}$, $Xa_2 \subseteq Xb_{i_2}$, ..., $Xa_n \subseteq Xb_{i_n}$ et que $t_{b_j} - tb_1$ est égal à $ta_j - ta_1$ pour tout $j \in 1..n$. La *fréquence relative* ou *fréquence* d'une séquence sa est définie comme le pourcentage de séquences $s \subseteq D$ tel que $sa \subseteq s$, et est notée $supD(sa)$ ou simplement $sup(sa)$ lorsque le contexte est clair.

Le problème général de *la recherche de motifs séquentiels avec des contraintes temporelles* [87] est de découvrir toutes les séquences s d'une base de séquences temporelles D telles que $supD(s) \geq minsup$ et que s respecte toutes les contraintes temporelles. Quatre types de contraintes temporelles sont proposées par Hirate et Yamana [87] :

- C1 : l'intervalle de temps minimal entre deux itemsets adjacents d'un motif séquentiel,
- C2 : l'intervalle de temps maximal entre deux itemsets adjacents d'un motif séquentiel,
- C3 : la durée minimale en temps d'un motif séquentiel,
- C4 : la durée maximale en temps d'un motif séquentiel.

Ces contraintes sont une généralisation des contraintes temporelles utilisées dans les autres algorithmes de recherche de motifs séquentiels [87]. Pour le problème général de la recherche de motifs séquentiels avec des contraintes temporelles, Hirate et Yamana [87] ont proposé un algorithme qui est une extension de PrefixSpan. Cette thèse présente cet algorithme, avec quelques légères modifications, car il est la base de l'algorithme présenté dans ce chapitre.

4.2.2 L'algorithme de Hirate et Yamana

L'algorithme de Hirate et Yamana effectue une recherche en profondeur. Comme PrefixSpan il est basé sur la propriété que si un motif séquentiel n'est pas fréquent, les motifs séquentiels le contenant ne seront pas fréquents. Cette propriété permet de trouver tous les motifs fréquents en trouvant d'abord les motifs fréquents contenant un seul item, puis de prendre chacun de ces motifs pour les étendre récursivement en ajoutant un item à la fois à sa droite pour tenter de trouver de nouveaux motifs fréquents, mais en n'étendant pas les motifs non fréquents.

L'algorithme procède en projetant de façon récursive une base de séquences en un ensemble de bases de séquences contenant un nombre de séquences inférieur ou égal. Ce processus facilite la construction des motifs séquentiels un item à la fois. Dans la suite, la notation $ISDB|(t, i)$ représente l'opération de projeter une base de séquences temporelles $ISDB$ avec une paire comprenant une estampille temporelle t et un item i . La figure 4.2 illustre l'algorithme pour calculer la projection $ISDB|(t, i)$. Lorsque n projections successives avec des paires $(t_1, i_1), (t_2, i_2), \dots, (t_n, i_n)$ sont appliquées à une base de séquences, la notation $ISDB| < (t_1, i_1), (t_2, i_2), \dots, (t_n, i_n) >$ est utilisée.

L'algorithme de Hirate et Yamana *algoHirate* et sa sous-routine *algoProjection* sont

calculerProjection($ISDB, t, i$)

1. $ISDB|(t, i) := \emptyset$.
2. POUR CHAQUE séquence $\sigma = \langle (t_1, X_1), (t_2, X_2) \dots (t_n, X_n) \rangle$ de $ISDB$.
 - (a) POUR CHAQUE itemset (t_x, X_x) de σ , tel que $i \in X_x$ et $t_x = t$.
 - i. SI $X_x \setminus \{i\} = \emptyset$ ALORS
 $s := \langle (t_{x+1} - t_x, a_{x+1}), \dots (t_n - t_x, X_n) \rangle$.
 - ii. SINON
 $s := \langle (0, X_x \setminus \{i\}), (t_{x+1} - t_x, a_{x+1}), \dots (t_n - t_x, X_n) \rangle$.
 - iii. SI $s \neq \emptyset$ et s satisfait $C1, C2$ et $C4$ ALORS
 $ISDB|(t, i) := ISDB|(t, i) \cup \{s\}$.
3. RETOURNER $ISDB|(t, i)$.

Figure 4.2 Algorithme pour calculer $ISDB|(t, i)$

algoHirate($ISDB, minsup, C1, C2, C3, C4$)

1. $R := \emptyset$.
2. Parcourir $ISDB$ pour trouver tous les items avec une fréquence supérieure ou égale à $minsup$.
3. POUR CHAQUE tel item fréquent i
 - (a) $R := R \cup \{ \langle (0, i) \rangle \}$.
 - (b) algoProjection($ISDB|(0, i), \langle (0, i) \rangle, R, minsup, C1, C2, C3, C4$).
4. RETOURNER R ;

Figure 4.3 Algorithme de Hirate et Yamana

algoProjection($ISDB, prefix, R, minsup, C1, C2, C3, C4$)

1. Parcourir $ISDB$ pour trouver toutes les paires (t, i) satisfaisant $minsup$ et tel que $t > C1$ et $t < C2$.
2. POUR CHAQUE paire (t, i) trouvée
 - (a) $newPrefix := Concatener(prefix, (t, i))$.
 - (b) SI $newPrefix$ satisfait $C3$ et $C4$ ALORS
 $R := R \cup \{newPrefix\}$.
 - (c) SI la taille de $ISDB|(t, i) \geq minsup$ ALORS
 algoProjection($ISDB|(t, i), newPrefix, R, minsup, C1, C2, C3, C4$).

Figure 4.4 Sous-routine *algoProjection* de l'algorithme de Hirate et Yamana

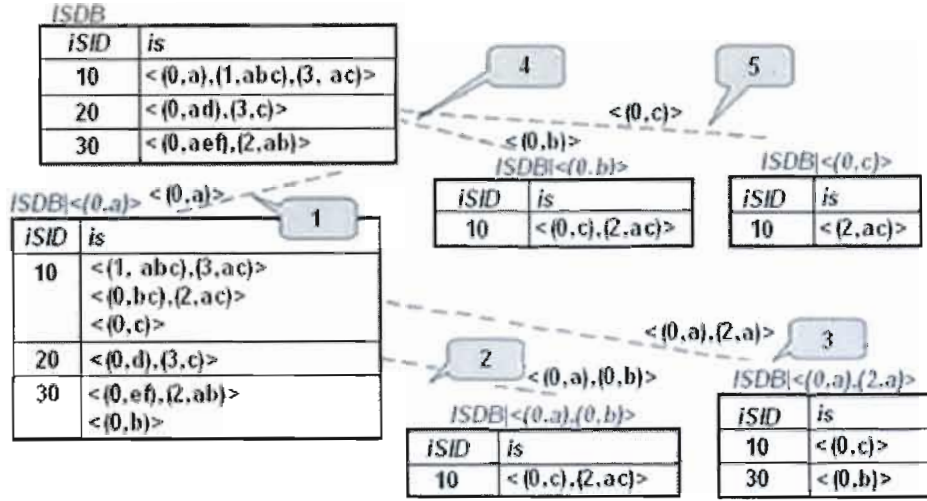


Figure 4.5 Une application de l'algorithme de Hirate et Yamana (adaptée de [87])

présentés à la figure 4.3 et 4.4, respectivement. Pour montrer le fonctionnement de cet algorithme, considérons son application pour la base de séquences *ISDB* illustrée à la figure 4.5 avec $minsup = 50\%$ et avec la contrainte $C2$ égale à 2 unités de temps. La base de séquences *ISDB* contient trois séquences nommées respectivement 10, 20 et 30. Dans cette base de séquences, les itemsets adjacents d'une séquence n'ont pas toujours un intervalle d'une unité de temps. Cela est pour illustrer l'algorithme de Hirate et Yamana pour le cas général, bien que dans cette thèse un intervalle d'une unité de temps est supposé être chaque itemset adjacent, comme expliqué précédemment.

Tout d'abord, *ISDB* est parcouru et les items fréquents a , b et c sont trouvés (instruction 2 de *algoHirate*). Par conséquent, $\langle 0, a \rangle$, $\langle 0, b \rangle$ et $\langle 0, c \rangle$ sont ajoutés à R , l'ensemble des motifs séquentiels trouvés (instruction 3.a de *algoHirate*). Puis, pour a , b et c , les bases de séquences *ISDB*| $\{0,a\}$, *ISDB*| $\{0,b\}$ et *ISDB*| $\{0,c\}$ sont créées respectivement. Pour chacune de ces bases de séquences, l'algorithme *algoProjection* est exécuté (instruction 3.b de *algoHirate*).

Ce dernier cherche tout d'abord toutes les paires fréquentes estampille et item qui vérifient les contraintes $C1$, $C2$ et $minsup$ (instruction 1 de *algoProjection*). Par exemple,

pour $ISDB|(0, a)$, les paires fréquentes $(0, b)$ et $(2, a)$ sont trouvées. Ces paires sont concaténées à $(0, a)$ pour obtenir les séquences $\langle (0, a), (0, b) \rangle$ et $\langle (0, a), (2, a) \rangle$, respectivement (instruction 2.a de *algoProjection*). Parce que ces motifs séquentiels respectent $C3$ et $C4$, ils sont ajoutés à l'ensemble R des motifs séquentiels trouvés (instruction 2.b de *algoProjection*). Puis, les bases de séquences projetées $ISDB| \langle (0, a), (0, b) \rangle$ et $ISDB| \langle (0, a), (2, a) \rangle$ sont calculées (instruction 2.c de *algoProjection*). Parce que ces bases de séquences contiennent plus de *minsup* séquences, *algoProjection* est exécuté (instruction 2.c de *algoProjection*). Après l'exécution complète de l'algorithme, l'ensemble de motifs séquentiels R contient les motifs $\langle (0, a) \rangle$, $\langle (0, a), (0, b) \rangle$, $\langle (0, a), (2, a) \rangle$, $\langle (0, b) \rangle$ et $\langle (0, c) \rangle$.

Deux remarques importantes doivent être mentionnées pour cet exemple. Tout d'abord, les itemsets d'un motif ayant la même estampille peuvent être fusionnés au fur et à mesure afin d'obtenir une représentation plus courte des motifs qui est équivalente. Ainsi le motif $\langle (0, a), (0, b) \rangle$ peut être réécrit $\langle (0, ab) \rangle$. Deuxièmement, au niveau de l'implémentation, la projection des bases de séquences par l'algorithme de Hirate et Yamana peut être effectuée par des *pseudo-projections*, c'est à dire que plutôt que de copier partiellement la base de séquences pour réaliser chaque projection, des pointeurs peuvent être utilisés sur la base de donnée originale. Cette stratégie proposée initialement dans PrefixSpan [134] améliore considérablement la performance de l'algorithme.

4.2.3 La performance de l'algorithme de Hirate et Yamana

L'algorithme de Hirate et Yamana est une extension de l'algorithme PrefixSpan, et en partage donc de nombreuses caractéristiques. Pour cette raison, il convient d'analyser tout d'abord la performance de l'algorithme PrefixSpan, puis ensuite d'analyser l'influence des changements apportés par Hirate et Yamana.

Dans l'article décrivant PrefixSpan [134], la complexité n'a pas été analysée formellement, mais la performance de l'algorithme a été comparée avec trois algorithmes de référence, SPADE [169], GSP [144] et FreeSpan [81], avec des jeux de données synthé-

tiques et réels. En occupation mémoire et en temps d'exécution, PrefixSpan a montré une performance supérieure. Depuis la publication de PrefixSpan, de nouveaux algorithmes plus performants ont été proposés comme PRISM [75] et LAPIN [166], mais reposant sur des mécanismes plus complexes comme l'encodage à base de nombres premiers. Bien que PrefixSpan ne soit le plus performant [75, 166], l'avantage de PrefixSpan est qu'il est facile à étendre (par exemple avec les contraintes temporelles) contrairement aux nouveaux algorithmes comme LAPIN et PRISM qui sont plus performants, mais qui n'ont pas d'extension. C'est pour cette raison qu'un algorithme basé sur PrefixSpan a été choisi pour ce travail.

Hirate et Yamana ont comparé la performance de leur algorithme avec celle de PrefixSpan [87] avec une base de séquences de mesures sismologiques. La conclusion de la comparaison est que leur algorithme est plus performant en temps d'exécution. Cela était toutefois prévisible, car ajouter des contraintes temporelles ne peut que réduire le nombre de motifs trouvés et l'espace de recherche. L'avantage sur PrefixSpan variera toutefois selon les contraintes utilisées : des contraintes plus strictes ou plus nombreuses pourront donner un plus grand avantage. Pour cette raison, il est impossible de quantifier cet écart ; il est propre à chaque cas (à l'extrême, C4, la durée maximale en temps d'un motif, pourrait être mise à zéro et ainsi seuls les motifs contenant un seul itemset seraient découverts). D'autre part, un autre facteur important est que l'algorithme de Hirate et Yamana utilise une définition différente de l'inclusion entre deux séquences, basée sur les estampilles temporelles. Cette définition permet de distinguer des motifs qui autrement seraient les mêmes, ce qui peut rendre des motifs non fréquents et diminue souvent considérablement le nombre de motifs trouvés et le temps d'exécution [87].

4.3 Extension de la recherche de motifs séquentiels avec le regroupement automatique de valeurs entières

Un deuxième problème rencontré en appliquant PrefixSpan pour extraire un espace problème est que les items sont des symboles indivisibles. Par conséquent, si certaines actions ont des paramètres, soit qu'elles sont encodées comme un seul item ou comme plusieurs

ID	Séquence temporelle avec valeurs		Motif fréquent	Fréq.
1	$\langle (0, a\{2\}), (1, b), (2, bc\{4\}) \rangle$	\rightarrow	$\langle (0, a\{2\}) \rangle$	33 %
2	$\langle (0, a\{2\}), (1, c\{5\}) \rangle$		$\langle (0, a\{5,5\}) \rangle$	33 %
3	$\langle (0, a\{5\}), (1, c\{6\}) \rangle$		$\langle (0, a\{2\}), (0, c\{5\}) \rangle$	33 %
4	$\langle (0, f), (1, a\{6\}) \rangle$		$\langle (0, c\{5\}) \rangle$	50 %
5	$\langle (0, f\ b\{3\}), (1, e), (2, f) \rangle$		$\langle (0, b\{2,5\}) \rangle$	33 %
6	$\langle (0, b\{2\}), (1, d\ f) \rangle$	

Figure 4.6 Exemple de recherche de motifs séquentiels avec regroupement de valeurs

items distincts correspondant à des catégories de valeurs de paramètres. Dans le premier cas, le problème est que l'information sur les paramètres est perdue. Dans le second cas, le problème est que la catégorisation est fixe. Par exemple, dans CanadarmTutor, les catégories de rotations de joint définies initialement pour les essais avec PrefixSpan étaient « petit », « moyen » et « grand », correspondant respectivement à des rotations de 0° à 60° , 60° à 100° et plus de 140° . Le désavantage de cette catégorisation est qu'elle est fixe, et donc que dans certaines situations les intervalles choisis ne distinguent pas de façon appropriée les actions réalisées par plusieurs opérateurs du bras canadien.

4.3.1 L'extension

Pour pallier cette limite, cette thèse propose (1) d'étendre la notion d'item dans une base de séquences pour que les items puissent avoir une valeur et (2) d'étendre l'algorithme de Hirate et Yamana pour catégoriser dynamiquement les items ayant des valeurs (l'extension est toutefois également applicable à PrefixSpan).

La thèse définit une *base de séquences temporelles avec valeurs* comme un ensemble de séquences temporelles où les items peuvent posséder une valeur. Une action a avec une valeur est notée $a\{valeur\}$. Dans le contexte de cette thèse, une valeur est un entier. Ce choix a été fait, car dans CanadarmTutor un entier est suffisant pour représenter les rotations de joint et les mouvements de caméra. Un exemple de base de séquences temporelles avec valeurs est illustré dans la partie gauche de la figure 4.6. Dans cette base, par exemple, les actions a et c de la séquence 1 ont les valeurs 2 et 4 respectivement.

Pour faire la recherche de motifs séquentiels dans une base de séquences temporelles avec valeurs, l'algorithme de Hirate-Yamana a été adapté. Le processus de comptage des item/paires (instructions 1 de *algoProjection* et instruction 2 de *algoHirate*) a été modifié pour noter les valeurs de chaque item compté ainsi que les identificateurs des séquences dans lesquelles ils se trouvent. Si la fréquence d'un item i (peu importe sa valeur) dans $ISDB$ est supérieur ou égal à $2 * minsup$, l'opération de projection appelle l'algorithme K-Means [106] avec l'ensemble des valeurs $V = v_1, v_2, v_3 \dots v_n$ trouvées pour cet item afin de créer des catégories (*clusters*) de valeurs similaires. La condition $2 * minsup$ est utilisée pour assurer qu'au moins deux groupes peuvent être formés avec une fréquence supérieure ou égale à $minsup$.

Pour être appliqué, l'algorithme K-Means prend en paramètre un nombre K de catégories à créer. Initialement, K-Means forme K groupes aléatoirement. Ensuite, il procède par itération. À chaque itération, il assigne chaque valeur de V au groupe ayant la moyenne la plus proche jusqu'à ce que tous les groupes restent les mêmes pour deux itérations successives.

Dans le cadre de l'algorithme de Hirate et Yamana modifié, K-Means est exécuté avec $K = 2$ en incrémentant K de un jusqu'à ce que le nombre de catégories avec une fréquence supérieure à $minsup$ n'augmente pas. Cet ensemble de groupes est conservé. Puis, les séquences de $ISDB|(t, i)$ sont séparées en une ou plusieurs bases de séquences selon ces groupes. Après, *algoProjection* est appelé pour chacune de ces bases de séquences avec une taille égale ou supérieure à $minsup$. Dans le cas où $ISDB|(t, i)$ est appelé à partir de *algoHirate* et que n groupes sont trouvés, plutôt que d'ajouter seulement $\langle (o, i) \rangle$ à l'ensemble R de séquences trouvés, $\langle (0, i\{v_{x1}\}) \rangle$, $\langle (0, i\{v_{x2}\}) \rangle$... $\langle (0, i\{v_{xn}\}) \rangle$ sont ajoutés, où $v_{x1}, v_{x2}, \dots v_{xn}$ sont les valeurs moyennes des groupes trouvés par K-Means. De façon similaire, *algoProjection* a été adapté pour que les motifs séquentiels soient construits en exécutant l'opération *Concatener* avec $(t, i\{v_{x1}\})$, $(t, i\{v_{x2}\})$... $(t, i\{v_{xn}\})$, au lieu de seulement $\langle (t, i) \rangle$. Ces traitements font que dans l'expansion d'un motif, les différentes catégories de valeurs pour un item sont traitées comme des items distincts.

La partie droite de la figure 4.6 montre des motifs séquentiels obtenus par l'exécution de l'algorithme modifié avec *minsup* égal à 33% (2 séquences) pour la base de séquences illustrée dans la partie gauche de la figure. Les séquences 1, 2, 3, 4 de cette base de séquences contiennent l'action a avec les valeurs $\{2, 2, 5, 6\}$ respectivement. Pendant la recherche de motifs, l'application de K-Means a créé les catégories $\{2, 2\}$ et $\{5, 6\}$ pour l'action a . Ces deux groupes ont respectivement les valeurs 2 et 5.5 comme valeur moyenne. Ainsi, les motifs $\langle (0, a\{2\}) \rangle$ et $\langle (0, a\{5.5\}) \rangle$ ont été identifiés comme des motifs distincts avec chacun une fréquence de 33 % plutôt qu'un seul motif avec une fréquence de 66%. L'avantage de cette approche est que les actions sont automatiquement groupées sur la base de la similarité de leur valeur, et que la moyenne est gardée comme indication des valeurs groupées. Un essai dans CanadarmTutor a montré que cette approche est bénéfique. Par exemple, cela a permis de distinguer deux motifs, l'un débutant par une rotation du joint EP avec une valeur moyenne de 15° , l'autre débutant par une rotation du même joint avec une valeur moyenne de 53° . Sans la catégorisation automatique, les deux motifs aurait été considérés comme débutant par la même action, soit « une petite rotation du joint EP ».

4.3.2 La performance de l'extension

Une question importante sur cette extension apportée à l'algorithme de Hirate et Yamana est son coût en termes de performance. Tout d'abord, le coût en mémoire supplémentaire est celui d'ajouter un entier à chaque item qui possède une valeur. Ce coût est négligeable. Au niveau du temps d'exécution, la complexité de l'algorithme K-means est très satisfaisante, car elle est linéaire. Plus précisément, la complexité est de $O(nkt)$ où n , k et t sont respectivement le nombre d'itérations, le nombre de groupes et le nombre d'objets à regrouper [80, p.403]. Bien que t dépend du jeu de données, les paramètres k et n peuvent être contrôlés. Dans CanadarmTutor, par exemple, une limite supérieure de 4 a été imposée respectivement à k et à n .

Dans l'extension proposée, K-Means a été choisi, car en l'appellant plusieurs fois avec des K variés, on peut contrôler le nombre de groupes formés, ce qui est pratique pour tenter

de trouver la partition offrant le plus grand nombre de groupes fréquents. Toutefois, il serait possible d'adapter l'extension proposée pour utiliser un autre algorithme que K-Means ou bien une variante de K-Means. Par exemple, PAM, une variation de K-Means qui utilise la valeur médiane plutôt que la valeur moyenne aurait pu être employée, afin que K-Means soit moins sensible aux valeurs extrêmes [80, p.404]. La complexité de PAM est de $O(k(n - k)^2)$ [80, p.406].

4.4 Extension de l'algorithme de Hirate-Yamana pour extraire la représentation compacte des séquences fermées

Une troisième limite est que les motifs séquentiels extraits par PrefixSpan ou par l'algorithme de Hirate et Yamana peuvent contenir beaucoup de redondance. Par exemple, ce dernier pourrait trouver les motifs fréquents $\langle (0, a) \rangle$, $\langle (0, a), (1, b) \rangle$ et $\langle (0, a), (1, b), (2, c) \rangle$ dans une même base de séquences. En fait, pour un motif séquentiel fréquent, tous les sous-motifs le composant sont également fréquents. Dans le pire cas, lorsqu'un motif fréquent contient n éléments qui sont distincts, il y a $2^n - 1$ motifs fréquents inclus dans celui-ci en excluant la séquence vide. Par exemple, un motif de 25 éléments distincts contiendra 33554431 motifs, alors qu'un motif de 50 éléments distincts contiendra environ 10^{15} motifs. Ainsi, plus les motifs trouvés sont longs, plus le nombre de motifs fréquents peut être grand. Dans le contexte de cette thèse où il est désiré d'extraire un modèle de tâche, seuls les motifs *fermés* ou *maximaux* pourraient en être conservés pour réduire le nombre de motifs à considérer par les services tutoriels.

Les *motifs séquentiels fermés* sont les motifs qui ne sont pas contenus dans un autre motif ayant la même fréquence [162]. Chaque motif fermé est l'élément maximal d'une classe d'équivalence de motifs partageant la même fréquence, ou plus précisément qui apparaissent dans exactement les mêmes séquences. Ces motifs sont partiellement ordonnés par la relation d'inclusion. Les plus petits éléments d'une classe d'équivalence sont appelés *générateurs minimaux* et l'élément maximal unique, *motif fermé*. Les *motifs séquentiels maximaux* sont les motifs séquentiels qui ne sont pas contenus dans aucun autre motif. Ils sont un sous-ensemble des motifs fermés.

ID	Séquence	→	ID	Motif	Fréquence
S1	a, b, e		M1	a, e	83 %
S2	a, d, e		M2	b, e	50 %
S3	e, f, g		M3	a, d, e	50 %
S4	b, a, d, e		M4	f, g	33 %
S5	{a, b}, f, g, e		M5	e	100 %
S6	a, d, e		...		

Figure 4.7 Exemple de recherche de motifs séquentiels fermés

Dans ce travail, l'algorithme de Hirate et Yamana a été adapté pour extraire les motifs fermés plutôt que les motifs maximaux. La raison est que l'ensemble des motifs fermés permet de reconstituer l'ensemble de tous les motifs séquentiels ainsi que leur fréquence [162]. Les motifs fermés sont donc une représentation compacte sans perte d'information de l'ensemble des motifs séquentiels. Ceci n'est pas vrai pour les motifs maximaux qui peuvent permettre de reconstruire l'ensemble des motifs fréquents, mais sans restituer leur fréquence.

Un exemple d'extraction de motifs fermés à partir d'une base de séquences sans information temporelle avec $minsup = 33\%$ est présenté à la figure 4.7. Pour extraire des motifs fermés à partir d'une base de séquences temporelles, l'algorithme de Hirate-Yamana a été modifié pour intégrer le « Bi-Directional Extension closure checking » (BIDE) et « BackScan pruning » (BackScan) de l'algorithme BIDE+ [162]. BIDE+ est une extension de PrefixSpan pour vérifier si un motif séquentiel est fermé lors de sa découverte sans avoir à le comparer à d'autres motifs. Il a été proposé comme une extension de PrefixSpan, mais il peut-être adapté pour la recherche de motifs dans une base de séquences temporelles avec l'algorithme de Hirate et Yamana, car ce dernier est aussi basé sur le principe de découverte de motifs par projection de PrefixSpan.

BIDE vérifie si un motif est fermé en vérifiant dans les séquences de la base de séquences originale contenant le motif si un item existe avec la même fréquence et pourrait être ajouté au motif. Si tel est le cas, le motif n'est pas fermé (par définition, un motif

séquentiel fermé ne peut être inclus dans un autre motif ayant la même fréquence).

Dans l'implémentation de l'algorithme modifié, le BackScan Pruning de l'algorithme BIDE+ a aussi été implémenté. Il permet d'arrêter de construire des motifs qui sont garantis de ne pas produire de motifs séquentiels fermés. Le BackScan Pruning a l'avantage de rendre la recherche généralement beaucoup plus rapide que la recherche de tous les motifs séquentiels, puisqu'une partie de l'espace de recherche est ignorée [162].

Les deux prochaines sections présentent respectivement les mécanismes BIDE et le BackScan Pruning. Les exemples sont donnés avec des séquences régulières (qui ne sont pas des séquences temporelles), car BIDE+ est conçu pour être appliqué à des bases de séquences plutôt que des bases de séquences temporelles. Ensuite, la section suivante décrit les ajustements pour intégrer le BIDE dans l'algorithme d'Hirate et Yamana. Cette intégration est simple à réaliser, et a été effectuée indépendamment par Lin qui l'a décrite avec des variations mineures dans un article d'atelier en 2006 [102] et dans un article de revue en 2008 [101].

4.4.1 Le BIDE

Le mécanisme BIDE [162] est basé sur l'idée qu'un motif est fermé s'il n'existe pas un item pouvant lui être ajouté tel que le motif résultant a la même fréquence.

Pour vérifier si un item peut être ajouté à un motif, le mécanisme BIDE vérifie tout d'abord si un item peut être ajouté après le dernier item de la séquence. Puisque les algorithmes basés sur PrefixSpan construisent les motifs récursivement en ajoutant un item à la fois après le dernier item de la séquence, cette vérification peut-être faite simplement en vérifiant si la prochaine récursion permet d'obtenir un motif ayant la même fréquence [162]. Si oui, le motif n'est pas fermé. Cette vérification a été nommée vérification de la « forward-extension » [162].

La deuxième vérification faite par BIDE est celle de la « backward-extension » et est la plus complexe. Cette dernière consiste à vérifier si un item peut être ajouté à un

motif avant le dernier item du motif tel que le motif résultant a la même fréquence. Elle est donc complémentaire à la vérification de la « forward-extension ». Pour expliquer comment la vérification de la « backward-extension » est réalisée, quelques définitions préalables sont nécessaires. Elles sont offertes dans les prochains paragraphes.

La *première instance d'un motif s_1 dans une séquence s_2* est la sous-séquence de s_2 constituée du premier item de s_2 jusqu'au dernier item de la première occurrence de s_1 dans s_2 . Par exemple, la première instance du motif $\langle (a), (b) \rangle$ dans $\langle (c), (a), (bd), (c), (b) \rangle$ est $\langle (c), (a), (b) \rangle$. La *dernière instance d'un motif s_1 dans une séquence s_2* est la sous-séquence de s_2 constituée du premier item de s_2 jusqu'au dernier item de la dernière occurrence de s_1 dans s_2 . Par exemple, la dernière instance du motif $\langle (a), (b) \rangle$ dans $\langle (c), (a), (bd), (c), (b) \rangle$ est $\langle (c), (a), (bd), (c), (b) \rangle$.

Le *ième-dernier-dans-le-dernier* (noté LL_i) pour une séquence s_2 et un motif $s_1 \subseteq s_2$ de taille n , est définie comme suit. Si $i = n$, LL_i est la dernière occurrence du ième item de s_1 dans la dernière instance de s_1 dans s_2 . Si $1 \leq i < n$, LL_i est la dernière occurrence du ième item de s_1 dans la dernière instance de s_1 , et LL_i doit apparaître avant LL_{i+1} . Par exemple, pour $s_1 = \langle (a), (b) \rangle$ et $s_2 = \langle (a), (bd), (c), (b) \rangle$, LL_1 et LL_2 sont respectivement la première occurrence de a et la deuxième occurrence de b .

La *ième période maximale* pour une séquence s_2 et un motif $s_1 \subseteq s_2$ de taille n , est définie comme suit. Si $1 < i \leq n$, c'est la partie de la séquence s_2 entre le $i - 1$ ème élément de la première instance de s_1 et LL_i . Si $i = 1$, c'est la partie de la séquence s_2 située avant LL_1 . Par exemple, pour $s_1 = \langle (a), (b) \rangle$ et $s_2 = \langle (c), (a), (bd), (c), (b) \rangle$, la première et la deuxième période maximale sont respectivement $\langle (c) \rangle$, et $\langle (bd), (c) \rangle$.

La vérification de la « backward-extension » pour un motif m de taille n consiste à vérifier s'il existe un item apparaissant dans chaque ième période maximale pour les séquences qui contiennent m , pour un i tel que $1 \leq i \leq n$. Si un tel item existe, cela signifie qu'il permet d'étendre le motif m pour former un nouveau motif avec la même fréquence et donc que m n'est pas fermé (un motif fermé ne peut être inclus dans un autre motif

ayant la même fréquence). Pour une preuve détaillée, le lecteur peut se référer à l'article de Wang [162].

4.4.2 Le BackScan pruning

Le BackScan pruning utilise un principe similaire. Il utilise les deux notions suivantes (ici présentées avec séquences temporelles). Le *ième-dernier-dans-le-premier* (noté LF_i) pour une séquence s_2 et un motif $s_1 \subseteq s_2$ de taille n , est défini comme suit. Si $i = n$, LF_i est la dernière occurrence du i ème item de s_1 dans la première instance de s_1 dans s_2 . Si $1 \leq i < n$, LF_i est la dernière occurrence du i ème item de s_1 dans la première instance de s_1 , et LF_i doit apparaître avant LF_{i+1} . Par exemple, pour $s_1 = \langle (a), (b) \rangle$ et $s_2 = \langle (c), (a), (bd), (c), (b) \rangle$, LF_1 et LF_2 sont respectivement la première occurrence de a et la première occurrence de b .

La *ième période semi-maximale* pour une séquence s_2 et un motif $s_1 \subseteq s_2$ de taille n , est définie comme suit. Si $1 < i \leq n$, c'est la partie de la séquence s_2 entre le $i - 1$ ème élément de la première occurrence de s_1 et LF_i . Si $i = 1$, c'est la partie de la séquence s_2 située avant LF_1 . Par exemple, pour $s_1 = \langle (a), (b) \rangle$ et $s_2 = \langle (c), (a), (bd), (c), (b) \rangle$, la première et la deuxième période semi-maximale sont respectivement $\langle (c) \rangle$ et la séquence vide.

Le BackScan Pruning est un test effectué avant d'entreprendre chaque récursion avec un motif dans PrefixSpan. Il permet d'éviter d'explorer des motifs qui sont garantis de ne générer aucun motif fermé. Pour un motif m , le BackScan Pruning consiste à vérifier s'il existe un item apparaissant dans chaque i ème période semi-maximale pour les séquences contenant m , pour un i tel que $1 \leq i \leq n$. Si un tel item existe, cela signifie qu'il existe un item pouvant être ajouté à m avant le dernier élément de m pour obtenir un nouveau motif m' apparaissant dans les mêmes séquences. Puisque dans PrefixSpan, les motifs sont découverts en ajoutant récursivement des items après le dernier élément de chaque motif, il est certain que le motif m ne permettra pas de découvrir des motifs fermés, car toutes extensions de m seront nécessairement incluses dans une extension de m' qui

aura la même fréquence, puisque $m \subset m'$ et que m et m' apparaissent dans les mêmes séquences. Par conséquent, si le test est positif, il est garanti que m ne générera aucun motif fermé, et m ne doit pas être étendu. Pour une preuve plus détaillée, le lecteur peut se référer à l'article de Wang [162].

4.4.3 Intégration du BIDE et du BackScan Pruning à l'algorithme d'Hirate et Yamana

Pour intégrer BIDE et le BackScan Pruning dans l'algorithme d'Hirate et Yamana, il faut intégrer les contraintes temporelles et la notion d'intervalle de temps entre les items dans ces vérifications. Les prochains paragraphes expliquent comme cela peut être fait.

La vérification de la « forward-extension » pour un motif m consiste à vérifier si une récursion avec m résulte en un motif $m1$ ayant la même fréquence. Le seul changement est que $m1$ doit respecter $C1$, $C2$ et $C4$. La contrainte $C3$ n'a pas besoin d'être vérifié, car si m respecte $C3$, tout motif $m1 \supset m$ respectera également $C3$.

La vérification de la « backward-extension » pour un motif m de n items, consiste à vérifier s'il existe un item e apparaissant dans toutes les i ème périodes maximales des séquences contenant m pour un i tel que $1 \leq i \leq n$. Un premier changement à faire est que pour toutes les périodes, e doit avoir le même intervalle de temps avec l'item précédent la période et le même intervalle avec l'item suivant la période. De plus, ces intervalles doivent respecter $C1$ et $C2$. Il est inutile de vérifier $C3$, car si m respecte $C3$, tout motif $m1 \supset m$ respectera également $C3$. Pour la contrainte $C4$, une adaptation est nécessaire seulement pour $i = 1$, puisque c'est le cas de l'ajout d'un item e avant le premier item de m , ce qui peut augmenter la durée en temps de m . La modification à faire est de s'assurer que le motif résultant de la concaténation de e à m respecte $C4$.

Le test du « BackScan Pruning » pour un motif m consiste à vérifier s'il existe un item e apparaissant dans chaque i ème période semi-maximale des séquences contenant m pour un i tel que $1 \leq i \leq n$. Un premier changement à faire est que pour toutes les périodes, e doit avoir le même intervalle de temps avec l'item précédent la période et avoir le même

intervalle pour l'item suivant la période. De plus, ces intervalles doivent respecter $C1$ et $C2$. Il est inutile de vérifier $C3$, car si m respecte $C3$, tout motif $m1 \supset m$ respectera également $C3$.

Toutefois, le « BackScan Pruning » n'est pas compatible avec la contrainte $C4$. La raison est que pour deux motifs m et m' respectant $C4$ tel que $m \subset m'$ et apparaissant dans les mêmes séquences, il n'est pas garanti que toutes les extensions par ajout après le dernier item de m seront incluses dans toutes les extensions du même type pour m' . Par exemple, si $m = \langle (0, a) \rangle$, $m' = \langle (0, c), (3, a) \rangle$ et que $C4 = 3$, le motif $\langle (0, a), (1, b) \rangle$ ne sera pas inclus dans un motif généré avec m' , car ce motif violerait $C4$. Ainsi, si la contrainte $C4$ est utilisé avec le « BackScan Pruning », il n'est plus garanti que tous les motifs fermés seront trouvés.

4.4.4 Compatibilité de BIDE+ avec le regroupement automatique des items avec valeurs

BIDE+ peut aussi ne pas générer l'ensemble des motifs fermés si le regroupement automatique des items avec valeurs similaires proposé à la section 4.3 est utilisé. La raison est que ce dernier mécanisme peut dynamiquement regrouper des items avec valeurs similaires et les traiter comme des items distincts, ce qui en change la fréquence dynamiquement, alors que la base de séquences que BIDE+ utilise pour faire les vérifications de la « backward-extension » et du « BackScan Pruning » demeure inchangée.

Pour faire face à cette limite, la solution suggérée par cette thèse est soit (1) de désactiver BIDE+, (2) de désactiver le regroupement automatique ou bien (3) d'accepter que tous les motifs fréquents fermés ne seront pas toujours découverts et que des motifs fréquents non fermés pourraient être trouvés. Dans CanadarmTutor, la troisième solution a été adoptée, car bien que l'ensemble des motifs peut ne pas être complet, tous les motifs trouvés sont des motifs apparaissant dans les données et le nombre de motifs trouvés dans les essais avec CanadarmTutor a été suffisamment grand pour offrir de l'aide aux apprenants dans de nombreuses situations (des détails sont donnés sur l'expérimentation

dans CanadarmTutor aux sections 4.9 et 5.7.2, et au chapitre 6). Évaluer avec une plus grande précision l'impact que peut avoir le regroupement automatique sur la recherche de motifs fermés est conservé pour l'instant comme un projet pour des travaux futurs.

Par ailleurs, une stratégie de découpage des séquences présentées à la section 4.6 rend les motifs plus courts et rend moins nécessaire d'extraire des motifs séquentiels fermés.

4.4.5 Performance de BIDE+

La performance de BIDE+ a été comparée avec celle d'un autre algorithme de recherche de motifs séquentiels fermés nommé CloSpan [165] dans un article des auteurs de BIDE+ [162]. La comparaison a été faite avec plusieurs jeux de données synthétiques et réels. Les conclusions sont que BIDE+ est généralement plus rapide et consomme moins de mémoire que CloSpan. Il a aussi été observé que le temps d'exécution et la consommation en mémoire de BIDE+ croît linéairement avec le nombre de séquences et qu'activer le « BackScan Pruning » peut réduire le temps d'exécution de plusieurs ordres de grandeur. Dans un autre article par les auteurs de BIDE+ [161], ce dernier a été comparé avec les algorithmes de recherche de motifs séquentiels non fermés PrefixSpan et Spade [134, 169]. Sans surprise, la performance de BIDE+ a été nettement supérieure vu que le nombre de motifs fermés fréquents est généralement largement inférieur à l'ensemble de tous les motifs fréquents.

4.5 Extension de la recherche de motifs séquentiels avec une information sur le contexte

Une quatrième limite à l'application de PrefixSpan pour l'extraction de modèles de tâche destinés à être utilisés dans des STI est qu'il ne tient pas compte du contexte de chaque séquence. Dans un système tutoriel, il serait utile par exemple, d'annoter les séquences avec des informations comme le niveau d'expertise ou des habiletés pour extraire des motifs contenant ces informations.

La solution adoptée pour pallier cette limite est d'ajouter des dimensions aux séquences.

ID	Dimensions					Séquence
	État solution	Expertise	Habilité1	Habilité2	Habilité3	
S1	succès	expert	oui	oui	oui	<(0,a),(1,bc)>
S2	succès	débutant	non	oui	non	<(0,d) >
S3	échec	expert	oui	oui	oui	<(0,a),(1,bc)>
S4	échec	intermédiaire	non	oui	oui	<(0,a),(1,c), (2,d)>
S5	succès	expert	non	non	oui	<(0,d), (1,c)>
S6	succès	débutant	non	non	oui	<(0,c), (1,d)

Figure 4.8 Exemple de base de séquences multidimensionnelles

ID	Dimensions					Séquence	Fréquence
	État solution	Expertise	Habilité1	Habilité2	Habilité3		
P1	*	expert	oui	oui	oui	<(0,a)>	33 %
P2	*	*	*	oui	oui	<(0,a)>	50 %
P3	*	expert	oui	oui	oui	<(0,a), (1,b)>	33 %
P4	succès	*	non	*	*	<(0,d)>	50 %
P5	succès	expert	*	*	oui	<(0,c)>	33 %
P6	succès	débutant	non	*	non	<(0,d)>	33 %

Figure 4.9 Exemple de motifs séquentiels multidimensionnels

Pinto et al. [136] a originalement proposé la recherche de motifs séquentiels avec dimensions, comme extension à la recherche de motifs séquentiels. Une *base de séquences multidimensionnelles* est définie comme un ensemble de séquences et un ensemble de n dimensions $E = E_1, E_2, \dots, E_n$, où chaque séquence (appelée *MD-séquence*) possède une valeur symbolique pour chaque dimension. Un ensemble de valeurs pour n dimensions est appelé un *MD-pattern* et est noté e_1, e_2, \dots, e_n .

La figure 4.8 illustre un exemple de base de MD-séquences. Il est important de noter que les séquences utilisées ici sont des séquences temporelles avec valeurs telles que définies dans les sections précédentes ce qui n'est pas le cas dans la définition de Pinto et al. [136] qui utilise des séquences régulières. La base de séquences illustrée comporte cinq dimensions. La première indique si une séquence d'actions à mené au succès ou à l'échec. La deuxième indique le niveau d'expertise de l'utilisateur ayant effectué une séquence d'actions. La troisième décrit les habiletés maitrisées par l'utilisateur. À titre d'exemple, dans la figure 4.8, la MD-séquence $S1$ est annotée avec les valeurs « succès » et « débutant » pour les dimensions « État solution » et « Expertise », respectivement, et les autres dimensions indiquent que l'utilisateur ayant effectué $S1$ possède les habiletés « Habileté1 », « Habileté2 » et « Habileté3 ». Ces dimensions sont des exemples de données qui peuvent annoter les séquences pour un contexte tutoriel. Il est néanmoins possible d'ajouter d'autres informations sur l'utilisateur telles que son âge, sa scolarité, ses préférences d'apprentissage et son état émotionnel, si les données sont disponibles.

Le symbole « * » qui subsume toutes les autres valeurs peut être utilisé comme valeur pour une dimension. Un MD-pattern $Px = ex_1, ex_2, \dots, ex_n$ est dit contenu dans un autre MD-pattern $Py = ey_1, ey_2, \dots, ey_n$, si pour tout $i \in \{1, 2, \dots, n\}$, $ey_i = *$ ou $ey_i = ex_i$. Une MD-séquence m_1 est dite contenue dans une autre MD-séquence m_2 si la séquence et le MD-pattern de m_1 sont contenus respectivement dans la séquence et le MD-pattern de m_2 . La fréquence relative d'une séquence (ou MD-pattern) pour une base de séquences multidimensionnelles D est définie comme le pourcentage de séquences (ou MD-patterns) la contenant. Similairement, la fréquence relative d'une MD-séquence pour une base de séquences multidimensionnelles D est définie comme le pourcentage de MD-Séquences

la contenant.

Le *problème de la recherche de motifs séquentiels multidimensionnels* est de trouver toutes les MD-séquences avec une fréquence supérieure ou égale à un seuil *minsup*. La table 4.9 illustre une partie des motifs séquentiels multidimensionnels obtenus à partir de la base de séquences multidimensionnelles de la table 4.8 avec *minsup* égal à 33% (deux MD-séquences). Le motif P3 représente de faire l'action *b* immédiatement après l'action *a*. Le motif P3 apparaît dans les séquences S1 et S3. Il a donc une fréquence de 33% ou deux MD-séquences. Parce que cette fréquence est supérieure ou égale à *minsup*, P3 est fréquent. De plus, le MD-pattern de ce motif indique que le motif a été effectué par un utilisateur débutant possédant les habiletés « Habileté1 », « Habileté2 » et « Habileté3 » et que P3 a été trouvé dans des séquences menant à l'échec et d'autres menant au succès.

4.5.1 Algorithmes de recherche de motifs multidimensionnels

Pinto et al. [136] ont proposé trois algorithmes pour la recherche de motifs multidimensionnels. Les deux premiers se nomment SeqDim et DimSeq. Ils sont basés sur la démonstration par Pinto et al. [136] que le problème de recherche de motifs multidimensionnels peut-être divisé en deux problèmes : (1) la recherche de MD-patterns fréquents et (2) la recherche de séquences fréquentes. La recherche de MD-patterns fréquents peut être effectuée avec un algorithme de *recherche d'itemsets fréquents* comme Apriori [2], FP-Growth [82] et Eclat [168], alors que la recherche de motifs séquentiels peut-être faite avec un algorithme de recherche de motifs séquentiels comme PrefixSpan. L'intérêt de cette division du problème de la recherche de motifs multidimensionnels dans le contexte de cette thèse est que l'algorithme de recherche de motifs séquentiels peut être l'algorithme de Hirate et Yamana étendu présenté dans les sections précédentes. Le troisième algorithme proposé par Pinto et al. [136] est basé sur une idée différente : il demande d'encoder les dimensions comme des actions dans les séquences pour ensuite appliquer un algorithme de recherche de motifs séquentiels. Il ne sera pas décrit, car il n'est pas compatible avec l'utilisation de contraintes temporelles et avec les mécanismes BIDE et

le Backscan Pruning puisqu'il demande d'encoder les valeurs de dimension comme des items dans un itemset supplémentaire dans les séquences.

Pour une base de séquences multidimensionnelle, l'algorithme SeqDim s'exécute de la façon suivante. Tout d'abord, les séquences fréquentes sont trouvées par un algorithme de recherche de motifs séquentiels en ne considérant que la partie séquence des MD-séquences. Puis pour chaque motif séquentiel trouvé, les MD-séquences qui le contiennent sont utilisées pour extraire des MD-patterns fréquents, qui sont ensuite combinés au motif séquentiel pour obtenir des MD-séquences fréquentes. Il est prouvé par Pinto et al. [136] que l'ensemble des MD-séquences générées ainsi constitue l'ensemble des MD-séquences fréquentes contenues dans une base de séquences multidimensionnelle.

L'algorithme DimSeq procède de façon inverse. Il extrait d'abord les MD-patterns fréquents. Puis pour chaque MD-pattern trouvé, les motifs séquentiels fréquents sont extraits des MD-séquences qui le contiennent, et ces motifs sont combinés au MD-pattern pour obtenir des MD-séquences fréquentes.

La complexité en temps de SeqDim et DimSeq est respectivement $\Theta(m + (p * n))$ et $\Theta(n + (p * m))$ où m , n et p sont le coût de la recherche des motifs séquentiels, le coût de la recherche des itemsets fréquents et le nombre de motifs trouvés dans la première étape [22].

Deux notes importantes sur l'implémentation de DimSeq et SeqDim sont les suivantes. Pour implémenter DimSeq, il faut choisir un algorithme d'extraction d'itemsets fréquents qui annote chaque motif avec la liste des séquences le contenant. Cela est le cas pour la plupart des algorithmes de recherche d'itemsets fréquents, mais ne l'est pas pour tous. Par exemple, pour l'algorithme Apriori [2], une variation nommée AprioriTid doit être utilisée [2]. Similairement, pour implémenter SeqDim, il faut choisir un algorithme de recherche de motifs séquentiels fréquents qui annote chaque motif avec la liste des séquences le contenant. Cela fonctionne bien avec les algorithmes basés sur PrefixSpan, mais aussi avec d'autres algorithmes comme Spade [169].

4.5.2 Recherche de motifs multidimensionnels fermés

Pour cette thèse, il a été choisi d'utiliser SeqDim avec l'algorithme de Hirate-Yamana étendu. SeqDim a été préféré à DimSeq car dans des tests préliminaires la recherche de motifs séquentiels était plus coûteuse en temps que la recherche de MD-patterns. Toutefois, rien n'empêche d'utiliser DimSeq.

Pour la recherche de MD-Patterns fréquents, l'algorithme Charm [168] a été implémenté. Cet algorithme a été choisi, car il est relativement simple à programmer, il est performant et il permet d'extraire l'ensemble des itemsets fermés fréquents, ce qui élimine une forme de redondance. Dans le contexte de la recherche de motifs séquentiels multidimensionnels, un itemset fermé fréquent est un MD-Pattern fermé fréquent. Les MD-patterns sont fermés s'ils ne sont pas inclus dans un autre MD-pattern ayant la même fréquence. Les MD-patterns fermés fréquents sont une représentation compacte et sans perte d'information de l'ensemble des MD-patterns fréquents [143]. Bien que Charm a été choisi pour découvrir les MD-Patterns fermés fréquents pour les raisons mentionnées, d'autres algorithmes de recherche d'itemsets fermés fréquents comme AprioriClose [132] pourraient le remplacer.

Il est important de noter que combiner la recherche de MD-patterns fermés avec la recherche de motifs séquentiels fermés avec SeqDim ou DimSeq ne résulte pas nécessairement en la découverte de l'ensemble des motifs multidimensionnels fermés. Ceci a été démontré par Songram [143]. Pour trouver l'ensemble des motifs multidimensionnels fermés, un post-traitement est nécessaire. Il consiste simplement à prendre chaque paire de motifs une à la fois pour éliminer les motifs inclus dans un autre motif ayant la même fréquence. Cette stratégie proposée par Songram [143] a été implanté pour cette thèse. Cette façon de procéder n'est toutefois pas très efficace, car pour $n > 1$ motifs trouvés, $n(n+1)/2$ comparaisons doivent être faites (ce qui a une croissance exponentielle), mais au moment de l'implémentation réalisée pour cette thèse, au début de 2008, c'était la seule façon que l'auteur de cette thèse a trouvée dans la littérature pour obtenir les motifs multidimensionnels fermés. Un algorithme amélioré a toutefois été présenté à la fin de

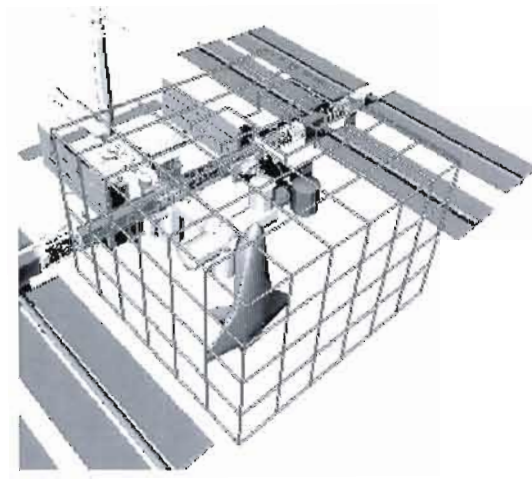


Figure 4.10 Division de l'espace 3D en cubes

2007 par Boonjing et Songram [22] qui permet d'éviter le post-traitement, et réduit aussi considérablement la complexité en temps. Utiliser cet algorithme pour l'implémentation réalisée pour cette thèse est une amélioration qui pourra être faite dans le futur.

4.6 Division des problèmes en sous-problèmes

Le dernier problème rencontré lors des premiers essais avec PrefixSpan découle du fait que les motifs séquentiels fréquents étaient extraits pour la totalité d'un exercice pouvant nécessiter l'application de plus d'une vingtaine d'actions. Hors, il a été observé dans des tests préliminaires qu'après environ six actions par un apprenant, il devenait parfois difficile pour le système de déterminer les motifs que l'apprenant suivait.

La solution à ce problème a été d'ajouter la notion d'*états de problème*. Par exemple, dans CanadarmTutor, où un exercice consiste à déplacer un bras robotisé pour atteindre une configuration particulière, l'espace 3D a été divisé en cubes, et un état de problème est défini comme l'ensemble des cubes contenant le bras robotisé à un instant donné. La division de l'espace en cubes a été faite sous forme de quadrillage avec 144 cubes de mêmes dimensions tel qu'illustré à la figure 4.10. Un état est défini seulement en fonction

des cubes et non en fonction des caméras choisies pour éviter que les états soient trop fins.

Un exercice dans CanadarmTutor est ensuite vu comme le passage d'un état initial P_1 à un état final P_f . Quand un apprenant effectue un exercice, deux types de séquences sont enregistrées : (1) la séquence d'états visités $A = \langle (0, P_1), (1, P_2), \dots (n, P_n) \rangle$ et (2) les séquences d'actions utilisées pour passer d'un état à l'état suivant (de P_1 à P_2 , de P_2 à P_3 , ..., de P_{n-1} à P_n).

Pour l'extraction des motifs séquentiels, l'algorithme de recherche de motifs séquentiels est appliqué (1) pour les séquences d'états visités et (2) pour chaque paire d'états afin de trouver les motifs fréquents pour passer du premier au deuxième. Diviser de long problèmes en sous-problèmes était nécessaire pour permettre un meilleur guidage des apprenants, parce qu'à tout moment, seuls les motifs pertinents pour l'état du problème sont à considérer.

4.7 Utilisation d'un minsup non constant

Une autre limite rencontrée provient du fait que le nombre de séquences contenues dans les bases de séquences traitées est petit initialement, quand peu de traces d'utilisation ont été enregistrées. Or, si *minsup* est choisi comme une constante, comme cela est proposée dans la littérature, tout ou presque tout est fréquent pour de petite bases de séquences. Par exemple, si *minsup* = 0.20, et qu'il y a de une à cinq séquences, en fréquence relative, *minsup* sera égal à une seule séquence. Autrement dit, tout motif apparaissant dans une seule séquence sera fréquent. Pour le contexte de cette thèse, il est préférable que *minsup* puisse être ajusté pour être plus haut lorsque le nombre de séquences est petit et que progressivement lorsque des séquences s'ajoutent, *minsup* diminue puis se stabilise, afin d'assurer que le nombre de séquences nécessaires pour qu'un motif soit fréquent soit élevé quand il y a peu de traces d'utilisation pour un problème et qu'ainsi l'intervalle de confiance ne soit pas trop grand.

La solution adoptée pour pallier ce problème est de spécifier *minsup* comme une fonction

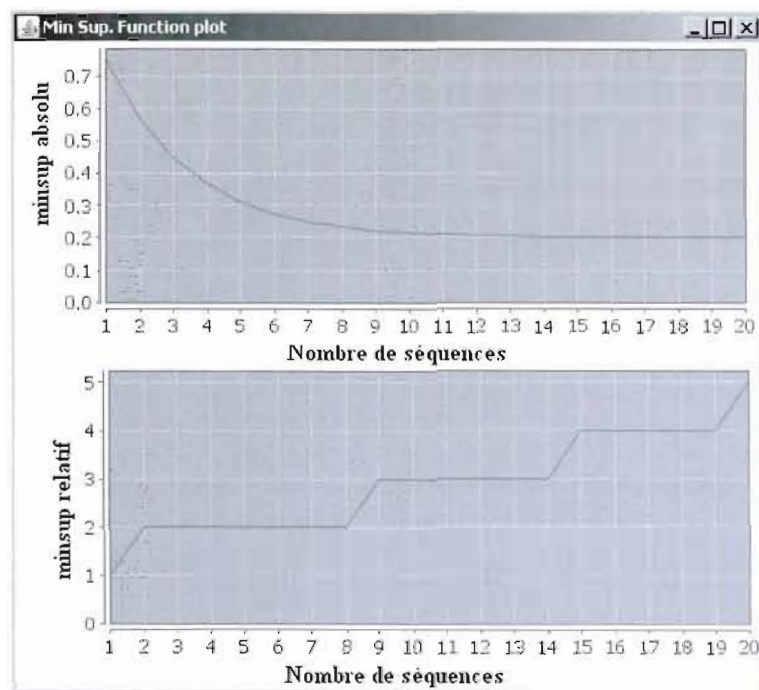


Figure 4.11 $minsupsup(x) = e^{(-0.4x-0.2)} + 0.2$

exponentielle de la forme $minsupsup(x) = e^{(-ax-b)} + c$ où x est le nombre de séquences contenues dans une base de séquences et a, b et c sont des constantes. Par exemple, choisir $minsupsup(x) = e^{(-0.4x-0.2)} + 0.2$ permet d'obtenir un $minsupsup$ élevé initialement puis un $minsupsup$ qui tend vers 0.20 lorsque le nombre de séquences augmente. Cette fonction est illustrée à la figure 4.11 en fréquence relative et en fréquence absolue pour $x = 1, 2, 3 \dots 20$. Pour tracer le graphique de la fréquence relative, le $minsupsup$ calculé est multiplié par x et la fonction partie entière par défaut (*floor*) est appliquée, ce qui donne $minsupsup(x) = \lfloor x \times e^{(-0.4x-0.2)} + 0.2 \rfloor$.

4.8 Les services tutoriels offerts sur la base des connaissances extraites

Les services tutoriels offerts dans CanadarmTutor sur la base des motifs séquentiels extraits sont présentement au nombre de trois et sont offerts dans les exercices procéduraux de déplacement du bras canadien. Ils sont tous basés sur l'utilisation d'un algorithme de

```

RecognizePlan(Trace_apprenant, Motifs)
1. Resultat :=  $\emptyset$ .
2. POUR CHAQUE motif M de Motifs.
   (a) SI Trace_apprenant  $\subseteq$  M.sequence ALORS
       i. Resultat := Resultat  $\cup$  {M}.
3. SI Resultat =  $\emptyset$  ET taille(Trace_apprenant)  $\geq$  2.
   (a) Retirer la dernière action de Trace_apprenant.
   (b) Resultat := RecognizePlan(Trace_apprenant, Motifs).
4. RETOURNER Resultat

```

Figure 4.12 Algorithme *RecognizePlan*

reconnaissance de plans nommé *RecognizePlan*. Cet algorithme est présenté à la figure 4.12.

L'algorithme *RecognizePlan* est exécuté après chaque action d'un apprenant. Il prend en paramètres la séquence d'actions effectuée par l'apprenant (*Trace_apprenant*) pour le problème courant ainsi qu'un ensemble de motifs fréquents en entrée. Quand l'algorithme est appelé pour la première fois, le paramètre *Motifs* est l'ensemble des motifs multidimensionnels. L'algorithme itère tout d'abord sur l'ensemble des motifs *Motifs* et note tous les motifs dont la partie séquence inclut *Trace_apprenant*. Si aucun motif n'est trouvé, l'algorithme enlève la dernière action (la plus ancienne) effectuée par l'apprenant de *Trace_apprenant* et cherche à nouveau des motifs s'appariant. La dernière action est enlevée, car les actions récentes sont jugées plus pertinentes que les anciennes pour un appariement. Le retrait d'actions est répété jusqu'à ce que l'ensemble des motifs s'appariant ne soit pas vide ou que la taille de *Trace_apprenant* soit inférieure à deux. Dans les tests effectués, retirer des actions a augmenté l'efficacité de l'algorithme de reconnaissance de plans de façon considérable, puisque cela rend l'algorithme plus flexible. De plus, une propriété intéressante de ce retrait progressif des actions est que si *Trace_apprenant* contient n actions, *RecognizePlan* cherchera tout d'abord les motifs où n actions s'appariant, puis $n - 1$, $n - 2$ et ainsi de suite. Cet ordre de recherche est bénéfique, car les motifs s'appariant davantage sont trouvés en premier.

Après que RecognizePlan a été exécuté, il est appelé les fois suivantes avec la nouvelle séquence *Trace_apprenant* et l'ensemble des motifs trouvés lors de la dernière exécution de RecognizePlan. Si cet ensemble est vide, RecognizePlan est appelé avec la totalité des motifs. Ceci permet à l'algorithme de ne pas considérer des motifs rejetés. Pour tirer profit de la notion d'état de problème, RecognizePlan a été modifié pour qu'à chaque moment seuls les motifs pertinents pour l'état courant soient considérés. Chaque fois que l'état change, RecognizePlan est appelé avec l'ensemble des motifs associés au nouvel état. De plus, à un niveau de granularité moins fin, le suivi des états visités par l'apprenant est aussi accompli par RecognizePlan. Cela permet de faire un lien entre les états. Les prochaines sous-sections décrivent les principaux services tutoriels qu'un agent tutoriel peut fournir sur la base de l'algorithme de reconnaissance de plan.

4.8.1 Évaluation du profil de l'apprenant

Le premier service tutoriel est l'évaluation du profil de l'apprenant en regardant les motifs que l'apprenant applique. Cela est fait en regardant les valeurs de dimension associées aux motifs suivis par l'apprenant, après chaque action faite par l'apprenant.

Dans CanadarmTutor, deux dimensions ont été initialement utilisées pour annoter les séquences d'états et les séquences d'actions (d'autres dimensions sont ajoutées au chapitre 6). La première est « niveau d'expertise » qui représente le niveau d'expertise de l'utilisateur et peut prendre trois valeurs : « débutant », « intermédiaire » et « expert ».

Dans CanadarmTutor, ce niveau d'expertise est fixé pour chaque utilisateur préalablement à l'utilisation du système et annote chaque trace produit par l'utilisateur. Dans des travaux futurs, le niveau d'expertise pourrait être défini comme pouvant varier au cours d'une session d'apprentissage en fonction de l'évaluation du modèle de l'apprenant. La deuxième dimension est « état de la solution » qui prend la valeur « succès » ou bien « échec » pour indiquer si l'exercice a été complété avec succès ou non par l'apprenant qui a enregistré la trace.

Pour chaque valeur de dimension, une entrée est créée dans le modèle de l'ap-

prenant avec un indicateur de maîtrise initialisée à 0.4 (le modèle de l'apprenant est le même que celui décrit à la section 2.7.2 pour le modèle cognitif). Pour l'évaluation du profil de l'apprenant, lorsque RecognizePlan détecte que l'apprenant suit un motif, l'indicateur pour chaque valeur de dimension associée au motif est augmentée de 0.01, et les indicateurs des valeurs alternatives à celles présentes dans le motif sont diminuée de 0.01. Par exemple, si RecognizePlan détecte que l'apprenant suit un motif avec la valeur « succès » pour la dimension « état de la solution », l'indicateur pour la valeur « succès » sera augmenté de 0.01, et celui associé à « échec » sera diminué de 0.01. Pour tous les indicateurs, un plafond et un plancher sont fixés à 0 et à 1, respectivement.

Ce mécanisme très simple, fait que CanadarmTutor peut estimer progressivement le profil de l'apprenant. Si par exemple, un apprenant applique 80% du temps des motifs avec la valeur « intermédiaire » pour la dimension « expertise », alors CanadarmTutor aura un indicateur de maîtrise élevé pour « intermédiaire » dans le modèle de l'apprenant et un indicateur bas pour « débutant » et « expert ».

L'évaluation réalisée de cette façon est une évaluation approximative, car si l'apprenant ne suit un motif que partiellement, son profil sera tout de même mis à jour. Une évaluation plus stricte aurait été de ne mettre le profil à jour que lorsqu'un motif est suivi complètement. Toutefois, pour un domaine comme la manipulation du bras canadien dans CanadarmTutor, cela donne de moins bons résultats, car un appariement complet est plus rare à cause du grand nombre de possibilités. Cela pourrait néanmoins être modifié pour d'autres domaines.

4.8.2 Suggestion d'indices personnalisés

Le deuxième service tutoriel est la suggestion d'indices personnalisés. Ce service consiste à déterminer les actions possibles à partir de l'état actuel et à proposer des actions à faire à l'apprenant. Dans CanadarmTutor, cette fonctionnalité est mise en oeuvre quand l'apprenant clique sur le bouton « What Next ? » de l'interface utilisateur pendant un exercice de déplacement du bras robotisé. L'assistance est alors présentée à l'apprenant

dans la fenêtre illustrée à la figure 4.13.

Dans la partie (A) de l'interface, les motifs d'états identifiés par RecognizePlan sont affichés sous forme de liste triée par la dimension « état de la solution » et la fréquence relative des motifs. Le premier motif illustrée dans la fenêtre à la figure 4.13 signifie qu'il a été accompli par des débutants qui ont réussi l'exercice, qu'il a une fréquence absolue de 5 (il est apparu dans 5 traces), et qu'il consiste à visiter les états suivants en ordre : 460, 462, 1367, 464, 465, 466.

Lorsque l'apprenant sélectionne un motif de la liste, une vue du premier état du motif est affichée dans la partie (B) de la fenêtre. Dans l'exemple de la figure 4.13, la vue de 460 est présentée par les caméras CP2, CP10 et CP8. Ces images peuvent être offertes, car elles sont sauvegardées automatiquement par CanadarmTutor pour chaque état quand un utilisateur enregistre une trace d'utilisation. Les caméras présentées sont celles que l'apprenant utilise au moment de faire appel au service tutoriel, ou si aucune image n'est disponible pour une de ces caméras, une caméra est choisie aléatoirement pour la remplacer parmi celles qui ont des images pour l'état. L'apprenant peut utiliser les boutons « flèche gauche » et « flèche droite » de la fenêtre pour visualiser les autres états (462, 1367, 464, 465, 466) du motif.

Dans la partie (C) de la fenêtre, les motifs d'actions sont présentés pour passer de l'état sélectionné à l'état suivant. Dans l'exemple, ces motifs sont les motifs d'actions identifiés par RecognizePlan pour passer de 460 à 462, car 460 a été sélectionné dans la partie (B) de l'interface. La liste présentant les motifs d'actions est triée par la dimension « état de la solution » et la fréquence des motifs. Dans l'exemple, le premier motif indique qu'il a été réalisé par des débutants qui ont réussi l'exercice et qu'il a une fréquence relative de 4. Lorsque le motif est sélectionné, les actions prescrites par ce motif sont présentées dans la partie (D) de l'interface. Dans cet exemple, le motif sélectionné consiste à décrémenter la valeur de rotation du joint SY de 10°, puis de décrémenter celle du joint EP de 20°.

Dans l'implémentation, il a été décidé de présenter tous les motifs à l'apprenant afin de le laisser choisir le motif qu'il préfère. Il est toutefois possible d'implémenter le service

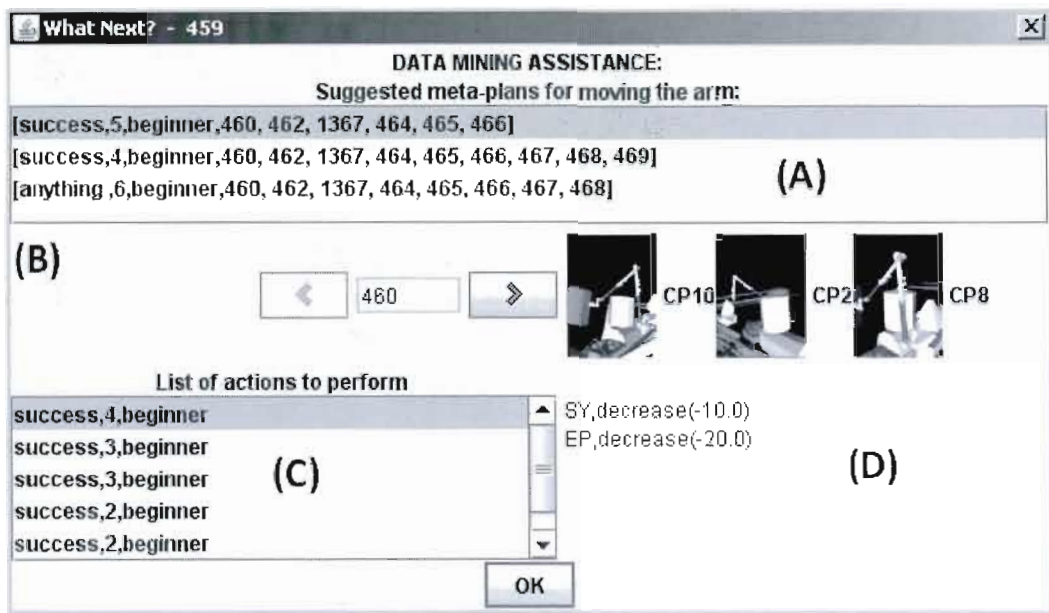


Figure 4.13 Suggestion d'incide par l'approche d'acquisition automatique

tutoriel autrement en présentant par exemple seulement le motif avec la fréquence la plus élevée dont les valeurs de dimensions s'apparient le mieux avec celles estimées par le modèle de l'apprenant. Une autre variante est de présenter des motifs correspondant à un niveau d'expertise légèrement supérieur à celui de l'apprenant pour l'aider à acquérir de nouvelles connaissances (par exemple, présenter des motifs de niveau "intermédiaire" aux débutants et des motifs de niveau "expert" aux intermédiaires). À un autre niveau, l'assistance du service tutoriel est offerte sur demande, mais il serait possible de le programmer pour intervenir quand RecognizePlan détecte que l'apprenant suit un motif menant à l'échec. Déterminer la meilleure approche pédagogique pour offrir des indices à l'apprenant est un travail qui a été conservé pour les travaux futurs.

4.8.3 Exploration libre des motifs

Le troisième service tutoriel est l'exploration libre des motifs pour un exercice. Ce service est accessible en cliquant sur le bouton « User solutions » de l'interface de CanadarmTutor, et est illustré à la figure 4.14. La fenêtre de gauche est celle qui s'ouvre initialement.

Dans la partie (A) de la fenêtre, l'ensemble des motifs d'états pour l'exercice courant est affiché. Tous les motifs sont préfixés par la valeur de la dimension « État de la solution », la fréquence absolue et la valeur pour la dimension « Niveau d'expertise ». Les motifs sont triés par la dimension « État de la solution » et la fréquence relative. Sous la liste illustrant les motifs, des filtres sont disponibles pour permettre de limiter l'affichage aux motifs de type « succès » ou « échec », ou de type « débutant », « intermédiaire » et « expert ». Deux cases à cocher permettent aussi de masquer les dimensions et la fréquence relative.

Lorsqu'un motif d'états est sélectionné dans la partie (A), une animation du motif est présentée dans la partie (B). Cette animation illustre la séquence d'états contenue dans le motif grâce aux captures d'écrans de chaque état. Seulement trois caméras sont affichées. Le but de l'animation est de donner à l'apprenant une vue d'ensemble d'un motif comme façon de résoudre l'exercice. Si l'apprenant désire avoir plus d'information sur le motif, il peut cliquer sur le bouton « Get details... » qui ouvre la fenêtre droite de la figure 4.14. Dans la partie (C) de la fenêtre, la liste des états du motif d'états sélectionné est présentée. Quand un état est sélectionné dans cette liste, les motifs d'actions pour passer de cet état au suivant sont présentés dans la liste (D) et des images de l'état sélectionné sont présentées en (E). Grâce à (C), (D) et (E), l'apprenant peut respectivement connaître les actions pour passer de chaque état au suivant tout en ayant une représentation visuelle de l'état sélectionné. Des fonctions de filtrage et de modification de l'affichage des motifs d'actions sont offertes sous la liste (D). Ce sont les fonctions présentée précédemment pour les motifs d'états.

4.9 Évaluation de la performance dans CanadarmTutor

Pour évaluer la performance de l'algorithme dans CanadarmTutor, nous avons appliqué l'algorithme à 170 traces représentant des solutions variées et longues pour un exercice de déplacement du bras robotisé dans CanadarmTutor. Un seul exercice a été utilisé, car l'extraction des motifs est faite séparément pour chaque exercice. Cet exercice est illustré à la figure 4.15. Il est un des exercices de déplacement parmi les plus longs

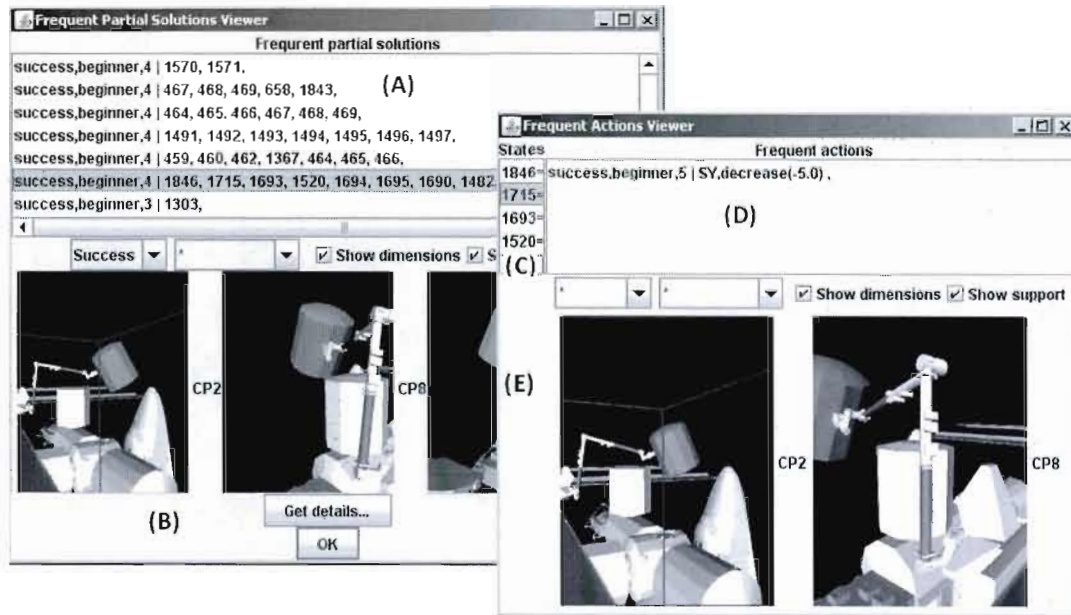


Figure 4.14 Exploration libre des motifs

et difficiles qui sont offerts dans CanadarmTutor. Il demande de contourner le module « JEMELM01 » de la station spatiale, de faire plusieurs changements/ajustements de caméras et d'utiliser plusieurs joints.

Pour cette expérience, il a été choisi d'utiliser des traces réelles plutôt que d'en générer aléatoirement parce générer des traces requiert de considérer de nombreux paramètres et que choisir une combinaison appropriée de ces paramètres pour avoir des données représentatives des données réelles n'est pas une tâche facile. Par exemple, des paramètres pertinents pour la génération aléatoire sont le nombre d'items, la probabilité que chaque item apparaisse dans une séquence, le nombre d'items et d'itemsets par séquence, la taille des itemsets, les valeurs associées aux items, la similarité entre les séquences, le nombre de motifs présents pour un *minSup* donné, les valeurs de dimensions pour chaque séquence et la façon dont le problème est divisé en sous-problèmes dans le contexte de CanadarmTutor. Utiliser des données réelles permet d'éviter la problématique de la génération de traces. Toutefois, la limite de l'utilisation de traces réelles est que le nombre de traces est forcément moindre.

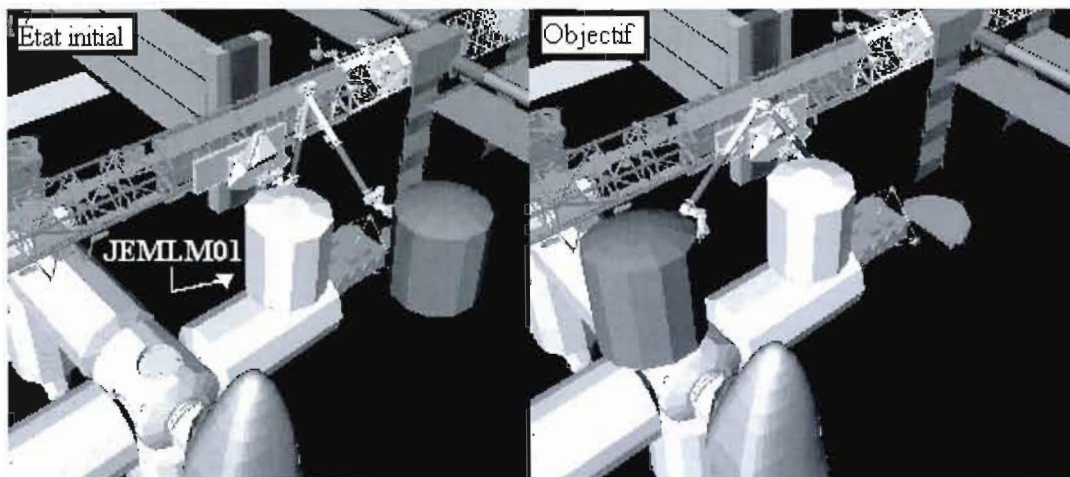


Figure 4.15 Un exercice de déplacement du bras robotisé

La prochaine sous-section décrit les caractéristiques des traces enregistrées. Ensuite, deux sous-sections décrivent respectivement les résultats de l'extraction de motifs d'états et de motifs d'actions à partir de ces traces. Puis la sous-section suivante offre une analyse et une discussion des résultats.

4.9.1 Caractéristiques des traces enregistrées

Les séquences d'états enregistrées contiennent en moyenne 47.9 états parmi un total de 1360 états. Chaque état était constitué en moyenne de 9.81 cubes avec un écart-type de 1.86 et un maximum de 14 parmi les 144 cubes divisant l'espace d'opération de CanadarmTutor. La figure 4.16 illustre le nombre de nouveaux états trouvés dans chaque nouvelle trace (non contenus dans les traces précédentes). La majorité des 1360 états ont été vus dans les cent dix premières traces pour la première fois, alors que peu de nouveaux états ont été vus dans les soixante-dix dernières. Ainsi, il semble que la plupart des états importants pour réussir l'exercice sont couverts par les traces.

Les séquences d'actions contenaient en moyenne 1,17 action avec un écart-type de 0.70 et un maximum de 14. Les séquences d'actions contiennent moins d'items que les séquences d'états à cause de la définition fine des états qui a été utilisé dans CanadarmTutor.

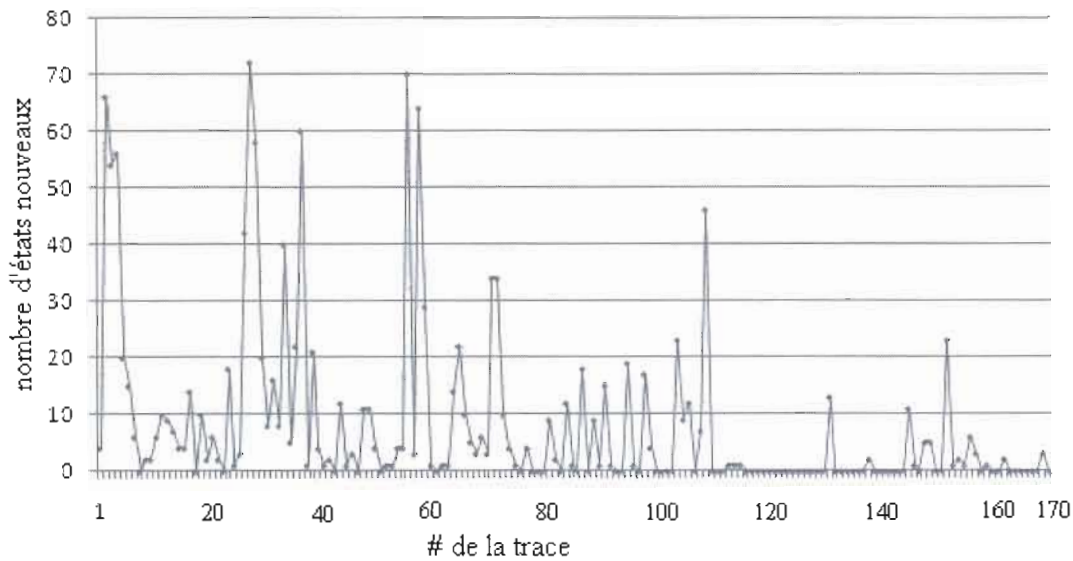


Figure 4.16 Nombre de nouveaux états pour chaque trace

Les dimensions employées pour annoter les séquences d'actions et d'états sont « niveau d'expertise » et « état de la solution » (cf. section 4.8.1) ainsi que 23 dimensions qui indiquent la maîtrise de connaissances importantes pour l'opération de Canadarm2 (ces dimensions seront justifiées au chapitre 6). Chaque séquence avait en moyenne 6.77 valeurs de dimension différentes de « * », avec un écart-type de 4.68 et un maximum de 13.

4.9.2 Résultats de l'extraction de motifs d'états

L'extraction de motifs d'états a été testée avec $minSup = 0.02$ (4 séquences), $C_1 = 0$, $C_2 = 1$ et $C_4 = \infty$ et la recherche de motifs multidimensionnels fermés. L'algorithme a pris 78,62 s et a trouvé 4667 motifs avec une moyenne de 18.45 états (le plus long contenant 161 états).

4.9.3 Résultats de l'extraction de motifs d'actions

L'extraction de motifs d'actions a été faite avec $\text{minSup}(x) = e^{(-0.65x-0.2)}+0.03$, $C_1 = 0$, $C_2 = 4$ et $C_4 = \infty$ et la recherche de motifs multidimensionnels fermés pour les 1360 paires d'états. Pour les raisons expliquées à la section 4.7, un minSup non constant a été utilisé. Le regroupement automatique des items avec valeurs a été activé avec $k = 3$ et $n = 3$ (cf. section 4.3).

L'algorithme a été exécuté une fois pour chacune des 2223 paires d'états. Chaque exécution a pris en moyenne 6.42 ms (un maximum de 1,07 s), pour un total de 14280 ms. Pour chaque paire d'état, il y avait en moyenne 3.66 séquences, avec un écart-type de 9.78 et un maximum de 142. En moyenne, 3.32 motifs ont été trouvés par paire (avec un écart-type de 10.08, et un maximum de 304), pour un total de 7388 motifs d'actions avec une taille moyenne de 26.74 actions. La fréquence minimum relative $\text{minSup}(x)$ était en moyenne 1.024 séquences avec un écart-type de 0.24 et un maximum de 5.

4.9.4 Discussion et analyse des résultats

Au cours de cette expérience, les temps d'exécution sont demeurés courts et satisfaisants (l'extraction de tous les motifs a pris à peine plus qu'une minute). Des paramètres jugés raisonnables ont été choisis pour l'algorithme. Les paramètres sont C_1 , C_2 , C_4 , n , k , minSup et la recherche de motifs fermés. Ajuster ces paramètres autrement (comme augmenter minSup) peut rendre l'algorithme plus efficace et réduire le nombre de motifs. De façon générale, une bonne combinaison de paramètres pour un algorithme de recherche de motifs séquentiels peut être trouvée en procédant par essais/erreurs en commençant par des paramètres restrictifs puis en passant à des paramètres de moins en moins restrictifs, ou inversement. Un travail de recherche intéressant pour des travaux futurs serait de trouver une manière d'automatiser la sélection des paramètres. Aussi, il est important de noter que l'implémentation de l'algorithme n'est pas entièrement optimisée. En faire une meilleure implémentation en améliorerait définitivement la performance.

Un autre aspect important pour la performance est la façon dont le problème est divisé en sous-problèmes (cf. section 4.6). Dans CanadarmTutor, un découpage très fin a été effectué. Par conséquent, les séquences d'actions sont courtes (en moyenne 2 actions), les séquences d'états sont longues (en moyenne 50 états), et il y a un très grand nombre d'états (1360 états). C'est pour cette raison que le temps pour trouver les motifs d'états (78,62 s pour une seule exécution) a été largement supérieur à celui de la découverte des motifs d'actions (14,28 s pour 2223 exécutions). Utiliser un découpage moins fin réduirait la taille des séquences d'états et ferait augmenter la taille des séquences d'actions, ce qui aurait pour impact de faire augmenter le temps pour l'extraction des motifs d'actions et de faire diminuer celui pour l'extraction des motifs d'états. Choisir un bon compromis entre ces deux temps d'exécution serait bénéfique. Dans CanadarmTutor, le découpage en sous-problèmes est un paramètre difficile à modifier, car il est profondément ancré dans l'architecture de CanadarmTutor. Pour cette raison et pour respecter les échéances du doctorat, le travail d'étudier l'impact de découpages moins fins dans CanadarmTutor a été conservé pour des travaux futurs.

Cette section ne s'est intéressé qu'à la performance des algorithmes dans le contexte de CanadarmTutor. L'appréciation des services tutoriels offerts avec les motifs découverts sera investiguée au chapitre 6.

4.10 Autres applications des algorithmes de motifs séquentiels dans le domaine de l'AIED

D'autres chercheurs ont appliqué la recherche de motifs séquentiels dans le domaine de l'apprentissage en ligne. Ces travaux peuvent être classés en deux groupes.

Le premier groupe sont les travaux pour la découverte de motifs qui sont interprétés par des humains. Par exemple, Kay et al. [135] ont extrait des motifs séquentiels à partir d'historiques d'interactions de travaux d'équipe; pour découvrir des motifs qui indiquerait de bons ou de mauvais comportements en apprentissage collaboratif, de façon à ce que des tuteurs humains puisse identifier les comportements problématiques dès le

début d'une session d'apprentissage. Similairement, Antunes [15] a extrait des motifs séquentiels pour identifier les déviations de comportement d'étudiants par rapport à un curriculum comprenant plusieurs cours dans le cadre d'un programme universitaire. Dans ce cas, une séquence est une liste de cours suivis par un étudiant. À partir de ces séquences, plusieurs motifs séquentiels ont été trouvés. Par exemple, un motif est que chaque fois qu'un étudiant échoue un cours au quatrième trimestre, il choisit un cours particulier d'économie à un trimestre suivant [15]. Romeo et al. [139] a proposé un outil de visualisation d'information qui a un but similaire. Il permet de visualiser les trajets suivis par des apprenants dans un curriculum. Plusieurs paramètres tels que la fréquence définissent les trajets affichés. L'approche a été appliquée dans un STI déployé sur le Web pour visualiser des motifs de navigation.

Un second groupe a utilisé la recherche de motifs séquentiels pour construire des systèmes recommandant des activités d'apprentissage aux apprenants. Par exemple, Su et al. [146] ont décrit un système qui suggère à un apprenant des activités en se basant sur ce que les apprenants avec un profil similaire ont fait. Leur approche consiste à (1) extraire des motifs séquentiels à partir de séquences d'activités d'apprentissage faites par des apprenants, (2) regrouper les étudiants qui ont démontré des motifs d'apprentissage similaires et (3) construire un arbre de décision pour chaque groupe à partir des motifs trouvés pour la recommandation d'activités. Un système avec un but similaire a été développé par Kristofic et Bielikova [99]. Ce système utilise les motifs trouvés dans les séquences de navigation d'apprenants pour recommander les concepts qu'un apprenant devrait étudier ensuite.

Tous ces travaux sont différents de l'approche d'acquisition automatique proposée dans ce chapitre, puisque leur but n'est pas d'extraire des connaissances du domaine. La réalisation est aussi différente.

4.11 Généricité de la solution

Bien que l'approche d'acquisition automatique présentée dans ce chapitre soit illustrée avec CanadarmTutor, elle est conçue pour être générique. Elle peut être réutilisée dans d'autres STI et l'algorithme de recherche de motifs séquentiels peut être employé pour d'autres tâches de fouille de données. Ces deux aspects sont discutés ci-après.

4.11.1 Généricité de l'approche pour d'autres STI

L'approche est conçue pour être appliquée pour des tâches procédurales et mal définies où il y a un grand nombre de possibilités et où il n'y a pas de stratégie claire pour trouver une solution. L'approche est basée sur le paradigme populaire en intelligence artificielle qui conçoit la résolution de problèmes comme l'application d'opérateurs ou d'actions atomiques pour passer d'un état initial à un état final [141], ce qui lui confère une grande généralité. Toutefois un certains nombre de contraintes doivent être respectées pour que l'approche soit applicable :

- l'état initial et les états finaux d'une tâche doivent être bien définis ;
- l'accomplissement d'une tâche doit pouvoir être représenté comme une séquence d'actions où les actions sont des symboles qui possèdent optionnellement une valeur ;
- les actions doivent être déterministes ;
- si les traces sont annotées, les annotations doivent pouvoir être représentées comme des symboles.

De nombreuses tâches peuvent répondre à ces contraintes. Un exemple est le diagnostic médical. Il serait possible de construire un simulateur où la tâche de faire un diagnostic consisterait à choisir une séquence de questions/tests pour atteindre un diagnostic valide. Le simulateur générerait automatiquement les résultats des questions/tests, mais ne connaîtrait pas la procédure du diagnostic. Des médecins devraient enregistrer des traces de tests/questions pour que le système apprenne des motifs qu'il pourrait ensuite utiliser pour conseiller des apprenants. Un deuxième exemple est la tâche de sélectionner

des actions militaires à entreprendre pour gagner une bataille simulée dans une guerre pour un stratège militaire. Un STI similaire à celui de l'exemple précédent pourrait être construit. Ces tâches sont des exemples de domaines où l'approche pourrait être applicable.

Toutefois, pour que l'application de l'approche soit un succès au niveau des services tutoriels offerts, il faut que le nombre de traces recueillies puisse couvrir l'espace des possibilités de façon satisfaisante pour la tâche. Si cela est difficile, une tâche peut être simplifiée pour réduire le nombre de possibilités, et un problème peut être divisé en sous-problèmes tel qu'expliqué dans ce chapitre.

Une autre obstacle potentiel à l'application de l'approche pour d'autres domaines est la performance de la recherche des motifs séquentiels. En particulier, le problème de la recherche de motifs séquentiels est plus difficile lorsque les séquences sont longues et que les séquences partagent plusieurs grandes sous-séquences. Pour adresser ce problème, des paramètres plus restrictifs peuvent être utilisés pour l'algorithme, par exemple, en augmentant *minSup*, en utilisant des contraintes temporelles plus contraignantes et en diminuant le nombre de dimensions annotant les motifs.

4.11.2 Généricité de l'algorithme en fouille de données

L'algorithme de recherche de motifs séquentiels étendu présenté dans ce chapitre peut être utilisé pour d'autres applications de fouille de données vu la forme générale des bases de séquences qu'il prend en entrée. Cette section présente une seconde application pour illustrer cette généricité. Comme cette autre application est une collaboration de l'auteur de cette thèse avec un autre étudiant dans son projet de doctorat, elle est décrite succinctement ici. Pour davantage de détails, le lecteur intéressé pourra se référer aux publications qui la décrivent [47, 49, 55, 59].

Le projet en question est un agent cognitif nommé CTS [41, 49] construit pour offrir une assistance à des apprenants lors d'activités d'apprentissage dans des environnements virtuels. CTS a aussi été intégré dans CanadarmTutor, mais pour tester des dialogues

tutoriels. Pendant des sessions d'apprentissage, CTS peut interagir avec des apprenants par des dialogues comprenant indices, questions et explications. CTS prend des décisions pédagogiques sur la base de règles encodées dans une structure nommée le *réseau des actes*, construite à la main par des experts [41, 49]. L'intégration de l'algorithme à CTS a été faite dans le cadre de l'ajout d'un mécanisme d'apprentissage épisodique à CTS [47]. L'objectif est de permettre à CTS d'apprendre de son comportement en réutilisant des séquences d'actions qui ont été fructueuses avec des apprenants. Le mécanisme fonctionne en trois étapes.

Enregistrement de traces : Après chaque exécution de CTS, une séquence comprenant ce que CTS a perçu ainsi que les actions prises par CTS est enregistrée dans une base de séquences. Les items d'une séquence sont des actions ou des perceptions et les valeurs associées sont des valences émotionnelles entre -1 et 1 attribuées par le mécanisme émotionnel de CTS [50].

Extraction de motifs séquentiels : Ce processus est répété après chaque exécution de CTS, dès sa cinquième exécution. Cela permet d'extraire des séquences fréquentes de perceptions et actions. Pour extraire les motifs, l'algorithme choisi est celui développé dans ce chapitre, car il permet la découverte de motifs en effectuant un regroupement automatique de valeurs similaires.

Exploitation des motifs pour adapter le comportement de CTS : À chaque cycle cognitif, CTS compare l'exécution courante avec les motifs extraits et choisit de suivre le motif le plus positif. Si aucun motif ne s'apparie, CTS prend une décision en s'appuyant sur ses autres mécanismes. Un motif séquentiel est dit positif, neutre ou négatif, si la somme des valeurs émotionnelles qu'il contient est respectivement supérieure à zéro, égale à zéro, ou inférieure à zéro.

Le mécanisme épisodique décrit ci-haut rend CTS capable de réutiliser des comportements l'ayant précédemment satisfait émotionnellement. Cela permet à un auteur d'encoder plusieurs comportements dans CTS et de le laisser apprendre lesquels sont les meilleurs. L'hypothèse est que la satisfaction de CTS résulte de l'apprentissage des apprenants. Des

expériences préliminaires réalisées avec le nouveau CTS ont montré que son comportement était amélioré [47]. Bien que le mécanisme proposé soit utilisé dans CTS, il aurait pu être utilisé dans d'autres agent intelligents où le comportement peut être enregistré sous forme de séquences et où une « fonction de satisfaction » peut-être exprimée. Le mécanisme proposé est une illustration d'une seconde application de l'algorithme étendu proposé dans ce chapitre.

4.12 Conclusion

Ce chapitre a présenté une approche d'acquisition automatique qui répond aux objectifs énoncés à la fin du chapitre précédent.

Le chapitre a d'abord abordé la problématique de la recherche de motifs temporels en fouille de données et a exposé le problème de la recherche de motifs séquentiels dans des traces d'utilisation. Ensuite, le chapitre a rapporté des essais préliminaires avec PrefixSpan dans CanadarmTutor qui ont permis d'identifier six problèmes à l'utilisation des algorithmes classiques de recherche de motifs séquentiels pour extraire des modèles de tâche partiel. Les sections 4.2 à 4.7 ont décrit ces problèmes en détails et proposé des solutions. En particulier, un algorithme étendu a été conçu, qui combine les propriétés de plusieurs autres algorithmes : (1) la recherche de motifs avec dimensions, (2) l'utilisation de contraintes temporelles, (3) la recherche de motifs fermés et (4) la catégorisation automatique d'items avec valeur. De plus, le chapitre a suggéré d'utiliser un *minsup* non constant et de diviser les problèmes en sous-problèmes.

Le chapitre a décrit trois services tutoriels qui sont offerts dans CanadarmTutor grâce aux motifs découverts : (1) l'évaluation du profil de l'apprenant, (2) la suggestion d'indices personnalisés et (3) l'exploration libre des motifs.

Une évaluation a été présentée avec CanadarmTutor. Finalement le chapitre a présenté une revue de littérature des autres applications de la recherche de motifs séquentiels dans le domaine de l'apprentissage en ligne, et a étudié la généricité de l'approche pour d'autres tâches et de l'algorithme développé en fouille de données.

CHAPITRE V

UNE EXTENSION DE L'APPROCHE AVEC DES RÈGLES SÉQUENTIELLES

Le chapitre précédent a présenté une approche pour l'acquisition automatique des connaissances du domaine dans les STI pour les domaines procéduraux et mal définis. L'approche est basée sur l'utilisation d'un algorithme de recherche de motifs séquentiels fréquents. Cet algorithme est une adaptation de l'algorithme PrefixSpan qui combine plusieurs extensions de la littérature et en ajoute aussi des originales.

L'approche d'acquisition automatique proposée repose sur l'idée de diviser un problème en sous-problèmes appelés *états* et d'extraire (1) des motifs d'états (des séquences fréquentes d'états visités) et (2) des motifs d'actions (des séquences fréquentes d'actions pour passer d'un état quelconque à un état adjacent). Cette division en sous-problèmes était nécessaire pour faciliter l'exploitation de l'espace des motifs comme par exemple, reconnaître le cheminement d'un apprenant dans cet espace.

Toutefois, une limite existe à cette solution : parce que les liens entre les motifs d'actions de différentes paires d'états ne sont pas pris en compte, il est impossible d'exploiter cette information pour offrir de meilleurs services tutoriels. Par exemple, l'approche présentée au chapitre 4, ne peut pas tenir compte que des apprenants ayant fait un motif m_x pour passer d'un état P_1 à P_2 ont majoritairement choisi ensuite d'effectuer le motif m_y lorsqu'ils arrivaient dans un état P_5 . Développer une méthode pour extraire et exploiter cette information améliorerait les services tutoriels offerts, en particulier

pour la suggestion des prochaines étapes à effectuer à un apprenant. Par exemple, si un apprenant effectue le motif m_x en P_1 , le service tutoriel pourrait lui suggérer le motif m_y , lorsqu'il arrivera en P_5 plutôt qu'un autre motif disponible pour P_5 .

Ce chapitre propose une solution à ce problème, soit une méthode pour permettre d'extraire des relations temporelles entre des motifs séquentiels et ainsi permettre de faire des prédictions sur les prochains motifs employés pour d'autres états. La prochaine section présente d'abord une revue de littérature des algorithmes utilisés en fouille de données pour extraire des relations temporelles pour la prédiction.

5.1 La problématique de la recherche de motifs temporels pour la prédiction

La découverte de relations temporelles entre des événements stockés dans de larges bases de données est importante pour plusieurs domaines parce qu'elle permet une meilleure compréhension des relations entre les événements, ce qui peut servir à la prédiction d'événements futurs. Par exemple, quelqu'un peut être intéressé à découvrir des relations temporelles entre l'appréciation des devises pour prendre des décisions d'achat ou de vente d'options sur le marché des options, dans le domaine du placement.

Dans le domaine de la fouille de données, une des techniques les plus populaires pour la découverte de relations temporelles entre des événements dans des séquences temporelles discrètes est la recherche de motifs séquentiels [4] (présentée à la section 4.1.1), qui consiste à trouver des séquences d'événements qui apparaissent fréquemment dans une base de séquences. Par contre, savoir qu'une séquence d'événements apparaît fréquemment n'est pas suffisant pour la prédiction d'événements. Par exemple, il est possible qu'un événement y apparaisse fréquemment après un événement x , mais qu'il y ait aussi plusieurs cas où x n'est pas suivi de y . Dans ce cas, prédire que y va arriver si x survient sur la base d'un motif séquentiel $\langle (x), (y) \rangle$ pourrait être une grande erreur. Pour la prédiction, il serait désirable d'avoir des motifs avec une mesure indiquant combien de fois x est apparu avant y et combien de fois x est apparu mais que y n'est pas apparu

ensuite. Ajouter cette information aux motifs séquentiels ne peut pas être fait facilement, parce que ces derniers sont des listes d'événements qui peuvent contenir plusieurs événements —pas seulement deux comme dans l'exemple précédent— et que les algorithmes proposés n'ont simplement pas été conçus pour permettre une telle extension.

5.2 La recherche de règles séquentielles : état de l'art

L'alternative à la recherche de motifs séquentiels pour la prédiction est *la recherche de règles séquentielles* [36, 37, 83, 79, 90, 107]. Une *règle séquentielle* (aussi appelée *règle épisodique*, *règle temporelle* ou *règle de prédiction*) indique que si un événement arrive, d'autres événements sont probables selon une certaine confiance ou probabilité. La recherche de règles séquentielles a été appliquée dans plusieurs domaines comme l'analyse des marchés boursiers [36, 90], la météorologie [79], l'analyse d'alarmes en télécommunication et la prévision des périodes de sécheresse [83, 37].

L'approche la plus populaire pour la découverte de règles séquentielles est celle de Manila [107] et de plusieurs autres chercheurs par la suite, qui vise à découvrir des ensembles d'événements partiellement ordonnés apparaissant fréquemment à l'intérieur d'une fenêtre de temps dans une séquence d'événements. Étant donné ces *épisodes fréquents*, un algorithme simple peut dériver des règles séquentielles respectant une confiance et une fréquence minimale. Ces règles sont de la forme $X \Rightarrow Y$ où X et Y sont deux ensembles d'événements, et sont interprétées comme « Si les événement(s) X surviennent, les événement(s) Y surviendront avec une certaine confiance ». Ces travaux sont conçus pour découvrir des règles apparaissant fréquemment dans une séquence d'événements. D'autres travaux pour extraire des règles séquentielles apparaissant dans une seule séquence d'événements sont les algorithmes de Hamilton et Karimi [79], Hsieh [90], Deogun et Jiang [37]. Le premier découvre des règles pouvant contenir plusieurs événements dans l'antécédent mais un seul dans le conséquent, le deuxième extrait des règles avec un seul événement dans l'antécédent et dans le conséquent, et le troisième, des règles sans limites sur le nombre d'événements dans l'antécédent et dans le conséquent.

Contrairement à ces travaux pour la découverte de règles dans une seule séquence d'événements, quelques travaux ont été faits pour découvrir des règles à partir d'un ensemble de séquences [36, 83, 104]. Le travail de Das et al. [36] permet de découvrir des règles où la partie gauche peut contenir plusieurs événements, mais où la partie droite est restreinte à un seul événement. Ceci est une limite importante, car dans des applications réelles, les relations séquentielles peuvent être entre plusieurs événements, et les mesures d'intérêt comme la confiance et la fréquence d'une règle utilisées par Das et al. [36] ne peuvent pas être déduites pour une règle à partir d'autres règles (par exemple, la confiance d'une règle $A \Rightarrow BC$ ne peut pas être déduite à partir de la confiance de règles $A \Rightarrow B$ et $A \Rightarrow C$). De plus, l'algorithme de Das et al. [36] est hautement inefficace puisqu'il teste toutes les règles possibles sans stratégie pour élaguer l'espace de recherche.

L'algorithme de Harms [83] découvre des règles séquentielles à partir de base de séquences sans restreindre le nombre d'événements contenus dans chaque règle. Cet algorithme cherche des règles avec une confiance et une fréquence supérieure ou égale à des seuils définis par l'utilisateur. La fréquence d'une règle est ici définie comme le nombre de fois que la partie droite apparaît après la partie gauche de la règle à l'intérieur d'une fenêtre de temps donnée (peu importe dans quelle séquence la partie gauche et la partie droite apparaissent).

Les algorithmes de Das et al. [36] et Harms [83] ont tous deux été conçus pour l'extraction de règles apparaissant fréquemment dans des séquences. Pour cette raison, ils sont inadéquats pour la découverte de règles communes à plusieurs séquences. Ceci peut être illustré par un exemple. Soit une base de séquences où chaque séquence correspond à un client, et chaque itemset d'une séquence représente les items achetés par ce client pendant un jour donné. Une personne pourrait être intéressée à extraire des règles séquentielles communes à plusieurs clients. Les algorithmes de Das et al. [36] et Harms [83] seraient inappropriés puisqu'une règle apparaissant plusieurs fois dans la même séquence pourrait avoir une fréquence élevée même si elle n'apparaît pas dans aucune autre séquence. Un deuxième exemple est la problématique de ce chapitre qui est de découvrir des règles séquentielles entre des motifs, communes à plusieurs tentatives de résolution

d'un problème pour lier les motifs d'actions pour différentes paires d'états. Pour cette application, il est crucial de trouver des règles communes à plusieurs séquences plutôt qu'apparaissant fréquemment dans une ou des séquences.

Seul Lo et al. [104] ont développé un algorithme pour extraire des règles communes à plusieurs séquences dans une base de séquences. Les parties gauche et droite de ces règles sont des séquences d'itemsets. Toutefois, une contrainte majeure de ce type de règle est que les items des parties gauches et des parties droites d'une règle doivent être ordonnés de la même façon dans les séquences pour qu'il puisse être comptabilisés pour la règle, et que même dans le cas le plus simple où une règle contient un seul itemset à gauche et un seul itemset à droite, les items de chaque itemset doivent être groupés en itemset de la même façon dans les séquences pour être comptabilisés pour la règle.

Pour le travail de cette thèse, cela est considéré comme une limite importante, car il est souhaité de trouver des règles entre plusieurs motifs d'actions dont les items de la partie gauche ou de la partie droite ne seront pas nécessairement toujours ordonnés de la même manière dans les séquences (la forme de ces règles sera précisée à la section 5.7). Par conséquent, plutôt que d'utiliser l'algorithme de Lo et al. [104], il a été décidé de concevoir de nouveaux algorithmes pour extraire une forme plus générale de règles séquentielles communes à plusieurs séquences. Pour ces règles, les parties gauche et droite sont des itemsets où les items ne sont pas obligés d'apparaître dans les mêmes itemsets dans les séquences d'une base de séquences. La seule contrainte est que les items de la partie gauche d'une règle doivent apparaître avant ceux de la partie droite. Cette approche permet de trouver des règles entre des items autrement introuvables avec l'algorithme de Lo et al. Ce chapitre présente deux algorithmes conçus pour extraire ce type de règles [69]. Parce que ces algorithmes ont une fin différente de ceux mentionnés dans cette sous-section, ils n'utilisent pas de fenêtre temporelle comme les algorithmes de Mannila et al. [107] et Harms et al. Harms [83].

Le premier algorithme procède en cherchant d'abord des règles d'association pour élarguer l'espace de recherche aux items apparaissant fréquemment ensemble dans plusieurs

séquences. Puis, il élimine les règles d'association qui ne respectent pas une confiance et une fréquence selon l'ordre temporel. Il sera démontré que procéder ainsi permet de découvrir toutes les règles séquentielles.

Le deuxième algorithme est une adaptation de l'algorithme de Deogun et Jiang [37] pour la découverte de règles communes à plusieurs séquences. Il découvre en premier lieu des règles entre deux items, puis combine les règles pour générer des règles contenant plus d'items. Cet algorithme utilise ainsi une génération de candidats par niveau semblable à celle de l'algorithme *Apriori* [2], un algorithme classique en fouille de données.

Ce chapitre est organisé de la façon suivante. La section 5.3 présente tout d'abord des définitions préliminaires. Ensuite, les sections 5.4 et 5.5 présentent respectivement le premier et le deuxième algorithme. Puis, une évaluation comparative des performances des algorithmes est présentée à la section 5.6. Après, la section 5.7 présente l'intégration des algorithmes à l'approche d'acquisition automatique présentée au chapitre précédent, et en particulier comment les services tutoriels de CanadarmTutor peuvent être enrichis en exploitant les règles séquentielles. Puis, la section 5.9 présente une seconde application des algorithmes développés. Finalement, la section 5.10 tire une conclusion.

5.3 Notions préliminaires

La *recherche de règles d'association* en fouille de données consiste à découvrir des associations entre des items dans une base de transactions. Formellement, une base de transactions E est définie comme un ensemble de transactions $T = \{t_1, t_2, \dots, t_n\}$ et un ensemble d'items $I = \{i_1, i_2, \dots, i_m\}$, tel que $t_1, t_2, \dots, t_n \subseteq I$. La *fréquence d'un itemset* $X \subseteq I$ pour une base de transactions est notée $sup(X)$ et est calculée comme le nombre de transactions qui contiennent X . Le problème de la recherche de règles d'association dans une base de transactions est de trouver toutes les règles d'association $X \rightarrow Y$ telles que $X, Y \subseteq I$ et $X \cap Y = \emptyset$, et que chaque règle possède au moins une valeur minimale pour certaines *mesures d'intérêt*. Les deux mesures d'intérêt initialement proposées [2] sont que les règles ont une fréquence et une confiance respectivement supérieure ou égale

ID	séquence		ID	règle	sup.	conf.
1	(a b), (c), (f), (g), (e)	→	1	$c \rightarrow a b e f$	0.5	1.0
2	(a d), (c), (b), (e f)		2	$a b c \rightarrow e$	0.5	1.0
3	(a), (b), (f) (e)		3	$a \rightarrow c e f$	0.5	0.6
4	(b), (f g)		4	$a b \rightarrow e f$	0.75	1.0
			5	$b \rightarrow e f$	0.75	0.75
			6	$a \rightarrow e f$	0.75	1.0
			7	$c \rightarrow f$	0.5	1.0
			8	$a \rightarrow b$	0.75	1.0
		

Figure 5.1 Un exemple d'extraction de règles d'association

à des seuils minimums $minSup$ et $minConf$ donnés. La *fréquence* d'une règle $X \rightarrow Y$ est définie comme $sup(X \rightarrow Y) = sup(X \cup Y)/|T|$. La *confiance* d'une règle est définie comme $conf(X \rightarrow Y) = sup(X \cup Y)/sup(X)$. Puisque $|T| \geq sup(X)$ pour tout $X \subseteq I$, la relation $conf(r) \geq sup(r)$ est vérifiée pour toute règle d'association r . Comme exemple de recherche de règles d'association, la partie gauche de la figure 5.1 illustre une base de transactions contenant quatre transactions et la partie droite présente des règles d'association trouvées avec $minSup = 0.50$ et $minConf = 0.50$.

Les règles d'association sont extraites à partir de bases de transactions. Une généralisation du concept de base de transactions est le concept de base de séquences [80]. Une base de séquences D est définie comme un ensemble de séquences $S = s_1, s_2, \dots, s_p$ et un ensemble d'items $I = \{i_1, i_2, \dots, i_m\}$ tel que chaque séquence s_x est une liste ordonnée de transactions $s_x = X_1, X_2, \dots, X_p$ tel que $X_1, X_2, \dots, X_p \subseteq I$.

Cette thèse définit le problème de la *recherche de règles séquentielles communes à plusieurs séquences* de la façon suivante. Une règle séquentielle $X \Rightarrow Y$ est une relation entre deux itemsets X, Y tel que $X, Y \subseteq I$ et $X \cap Y = \emptyset$. Cette définition est semblable à celle utilisée dans les travaux d'autres chercheurs tels que Deogun et Jiang [37] à l'exception de l'absence de contraintes temporelles et est identique à la définition de règle utilisée dans le domaine de la recherche de règles d'association [2]. L'interprétation d'une règle $X \Rightarrow Y$ est : si les items de X surviennent, les items de Y surviendront ensuite.

Il est important de noter qu'il n'y a pas d'ordonnancement entre les items dans X et entre ceux dans Y . Cette thèse définit deux mesures d'intérêt pour une règle séquentielle. Elles sont simplement une adaptation pour le cas de plusieurs séquences des mesures d'intérêts utilisées dans les travaux de Mannila [107], Das et al. [36] et Harms [83]. La première mesure est la *fréquence séquentielle d'une règle* et est définie comme $seqSup(X \Rightarrow Y) = sup(X \blacksquare Y) / |D|$. La deuxième mesure est la *confiance séquentielle d'une règle* et est définie comme $seqConf(X \Rightarrow Y) = sup(X \blacksquare Y) / sup(X)$. Ici, les notations $sup(X \blacksquare Y)$ et $sup(X)$ signifient respectivement le nombre de séquences d'une base de séquences où tous les items de Y apparaissent après tous les items de X , et le nombre de séquences contenant tous les items de X . Puisque $|D| \geq sup(X)$ est vrai pour tout $X \subseteq I$, la relation $seqConf(r) \geq seqSup(r)$ est vraie pour toute règle séquentielle r . Cette thèse définit le *problème de la recherche de règles séquentielles communes à plusieurs séquences* comme le problème de la découverte de toutes les règles séquentielles d'une base de séquences telles que leur fréquence et leur confiance sont respectivement supérieures ou égales à des seuils $minSeqSup$ et $minSeqConf$ donnés. Les règles respectant ces critères sont appelées *règles valides*. À titre d'exemple, la partie gauche de la figure 5.2 montre une base de séquences contenant quatre séquences et la partie droite présente des règles séquentielles trouvées pour $minSeqSup = 0.5$ et $minSeqConf = 0.5$.

Il est important de noter que le problème de recherche de règles séquentielles est largement différent de celui de la recherche de motifs séquentiels puisqu'il n'y a pas d'ordonnancement à l'intérieur des itemsets d'une règle. Par conséquent, il est impossible de simplement adapter les algorithmes de recherche de motifs séquentiels pour générer des règles séquentielles.

Le premier algorithme qui est proposé dans ce chapitre est basé sur l'observation que si l'on ignore ou enlève l'information temporelle d'une base de séquences D , on obtient une base de transactions D' . Pour chaque base de séquences D et sa base de transactions correspondante D' , chaque règle séquentielle $r : X \Rightarrow Y$ a une règle d'association correspondante $r' : X \rightarrow Y$ dans D' . Puisque $sup(X \blacksquare Y)$ est toujours inférieur ou égal à $sup(X \cup Y)$, les relations (1) $sup(r') \geq seqSup(r)$ et $conf(r') \geq seqConf(r)$ sont vraies

ID	séquence		ID	règle	seqSup	seqConf
1	(a b), (c), (f), (g), (e)	→	1	a b c \Rightarrow e	0.5	1.0
2	(a d), (c), (b), (e f)		2	a \Rightarrow c e f	0.5	0.66
3	(a), (b), (f) (e)		3	a b \Rightarrow e f	0.5	1.0
4	(b), (f g)		4	b \Rightarrow e f	0.75	0.75
			5	a \Rightarrow e f	0.75	1.0
			6	c \Rightarrow f	0.5	1.0
			7	a \Rightarrow b	0.5	0.66
		

Figure 5.2 Un exemple d'extraction de règles séquentielles

algoCMRules(D , $minSeqSup$, $minSeqConf$)

1. Obtenir la base de transactions D' en enlevant l'information temporelle de D .
2. Trouver toutes les règles d'association dans la base de transactions en appliquant un algorithme de recherche de règles d'association comme *Apriori* [2]. Choisir $minsup = minSeqSup$ et $minconf = minSeqConf$.
3. Parcourir la base de séquences D pour calculer la fréquence séquentielle et la confiance séquentielle de chaque règle d'association trouvée à l'étape précédente. Éliminer chaque règle r telle que $seqSup(r) < minSeqSup$ ou $seqConf(r) < minSeqConf$.
4. Retourner l'ensemble des règles restantes.

Figure 5.3 Algorithme CMRules pour l'extraction de règles séquentielles

pour chaque règle séquentielle r et sa règle d'association correspondante r' .

5.4 Algorithme CMRules pour la recherche de règles séquentielles communes à plusieurs séquences

Sur la base des observations de la relation entre les règles séquentielles et les règles d'association, cette thèse propose un premier algorithme pour la recherche de règles séquentielles. Cet algorithme est nommé CMRules [69] et est présenté à la figure 5.3. La figure 5.4 montre un exemple d'exécution de l'algorithme avec une base de séquences contenant quatre séquences.

L'algorithme prend en entrée une base de séquences D et les seuils $minSeqSup$ et $minSeqConf$, et produit en sortie l'ensemble des règles séquentielles contenues dans D qui respectent ces seuils. L'algorithme débute en ignorant l'information temporelle de

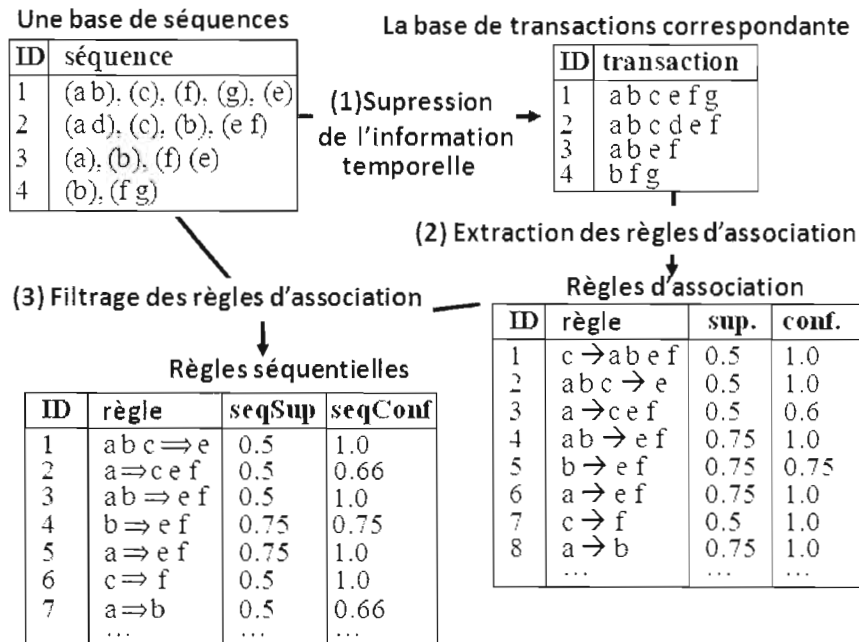


Figure 5.4 Un exemple d'exécution de l'algorithme CMRules

la base de séquences pour obtenir une base de transactions (étape 1 des figures 5.3 et 5.4). Il applique ensuite un algorithme de recherche de règles d'association sur la base de transactions avec les paramètres $minSup = minSeqSup$ et $minConf = minSeqConf$ (étape 2 des figures 5.3 et 5.4). Puis, l'algorithme calcule la fréquence séquentielle et la confiance séquentielle de chaque règle d'association en parcourant la base de séquences (étape 3 des figures 5.3 et 5.4). Il élimine ensuite les règles qui ne respectent pas les seuils $minSeqSup$ et $minSeqConf$ (étape 4 des figures 5.3 et 5.4). L'ensemble des règles restantes est l'ensemble des règles séquentielles.

5.4.1 Preuve de complétude

Cette sous-section démontre que l'algorithme trouve toutes les règles séquentielles. Pour le prouver, il faut démontrer que l'ensemble des règles d'association trouvées pour $minSup = minSeqSup$ et $minConf = minSeqConf$ contiennent l'ensemble des règles séquentielles respectant $minSeqSup$ et $minSeqConf$. La preuve suit.

Théorème 1 : L'algorithme découvre toutes les règles séquentielles pour des seuils $minSeqConf$ et $minSeqSup$ donnés, si $minConf \leq minSeqConf$ et $minSup \leq minSeqSup$.

Preuve du théorème 1 : Pour prouver le théorème, il est possible de considérer la fréquence séquentielle et la confiance séquentielle séparément, sans perte de généralité. Considérons la fréquence séquentielle. Supposons l'existence d'une règle r tel que $seqSup(r) \geq minSeqSup$ et $minSup > sup(r')$ (ce qui veut dire que la règle d'association correspondante n'est pas fréquente). Il a été montré précédemment que $sup(r') \geq seqSup(r)$. Par conséquent, $minSup > sup(r') \geq seqSup(r) \geq minSeqSup$. Donc, $minSup > minSeqSup$. Par conséquent, si $minSup$ est choisi tel que $minSeqSup \geq minSup$, il n'y aura pas de telle règle r .

Maintenant, considérons la confiance séquentielle. Soit une règle séquentielle r tel que $seqConf(r) \geq minSeqConf$ et que $minconf > conf(r')$. Il a été montré précédemment que $conf(r') \geq seqConf(r)$. Par conséquent, $minconf > conf(r') \geq seqConf(r) \geq minSeqConf$. Donc, $minconf > minSeqConf$. Par conséquent, si $minConf$ est choisi tel que $minSeqConf \geq minConf$, il n'y aura pas de telle règle r .

Corollaire 1 : L'algorithme est plus efficace si $minconf = minSeqConf$ et $minSup = minSeqSup$.

Preuve du corollaire 1 : Pour la recherche de règles d'association, plus $minSup$ et $minConf$ sont petits, plus le nombre de règles trouvées et le temps de recherche peut augmenter. Par conséquent, pour l'algorithme CMRules, il est préférable de sélectionner la plus grande valeur possible pour $minSup$ et $minConf$ qui va découvrir toutes les règles d'association contenant les règles séquentielles désirées. Le théorème 1 indique que ces valeurs sont bornées par $minSeqConf$ et $minSeqSup$ pour avoir une garantie que toutes les règles séquentielles seront obtenues. Par conséquent, les plus grandes valeurs que quelqu'un peut donner à $minConf$ et $minSup$ sont respectivement $minSeqConf$ et $minSeqSup$.

5.4.2 Implémentation efficace de l'étape 3 de l'algorithme

Cette section décrit comment implémenter l'étape 3 de l'algorithme de façon efficace. L'approche naïve serait de consulter chaque séquence de la base de séquences pour vérifier si chaque règle y apparaît pour calculer la valeur $sup(X \blacksquare Y)$ nécessaire pour calculer la confiance séquentielle et la fréquence séquentielle. Vérifier si une règle est contenue dans une séquence peut être fait en temps linéaire (une séquence peut être parcouru une seule fois du début à la fin pour voir si X apparaît et X est suivi par Y). Néanmoins, vérifier si une règle est contenue dans chaque séquence est inefficace. Les prochains paragraphes décrivent comment minimiser le nombre de séquences à vérifier pour chaque règle, mais cette explication requiert d'abord d'expliquer le processus d'extraction des règles d'association.

Pour découvrir les règles d'association présentes dans une base de transactions, un algorithme tel qu'Apriori peut être appliqué [2]. Il est important de noter que la thèse réfère ici à l'algorithme d'extraction de règles d'association Apriori (ce dernier utilise un algorithme d'extraction d'itemsets fréquents qui est aussi appelé Apriori dans la littérature). Les algorithmes de générations de règles d'association procèdent en deux phases [2]. Ils découvrent les itemsets fréquents, puis les utilisent pour générer les règles d'association. Un itemset fréquent est un itemset qui apparaît dans plus de $minSup$ transactions d'une base de transactions. La *fréquence d'un itemset* est définie comme le nombre de transactions qui le contiennent. Générer les règles d'association est fait en sélectionnant des paires d'itemsets fréquents X et Y tel que $X \subseteq Y$, pour générer des règles de la forme $X \rightarrow Y \setminus X$ ou bien $Y \setminus X \rightarrow X$ (voir [2] pour un algorithme rapide).

Pour implémenter efficacement l'étape 3 de l'algorithme, il faut modifier l'algorithme de recherche d'itemsets fréquents de façon à ce que chaque itemset X trouvé soit annoté avec l'ensemble des transactions qui le contiennent. Ceci est une modification triviale pour un algorithme tel que Apriori (dans le cas d'Apriori, cette variation se nomme *Apriori-Tid* [2]), et d'autres algorithmes le font par défaut (par exemple Eclat [168]). Avec cette information supplémentaire sur les itemsets fréquents, l'étape 3 peut être

effectuée efficacement en vérifiant chaque règle $X \rightarrow Y$ seulement avec les séquences qui contiennent son itemset X . Ceci permet de calculer correctement la fréquence séquentielle et la confiance séquentielle d'une règle, puisque $\text{sup}(X \blacksquare Y)$ peut être calculé en parcourant seulement les séquences contenant X (de par la définition de $\text{sup}(X \blacksquare Y)$), et que les autres termes pour calculer la fréquence séquentielle et la confiance séquentielle ($\text{sup}(X)$ et $|D|$) sont connus. Dans les tests effectués, cette simple optimisation a rendu l'étape 3 jusqu'à 50 % plus rapide.

5.4.3 Fusion des étapes 2 et 3 de l'algorithme

Il est possible de fusionner les étapes 2 et 3 pour que chaque règle d'association qui est générée soit immédiatement vérifiée pour sa fréquence séquentielle et sa confiance séquentielle. Ceci permet de considérer seulement une règle d'association à la fois en mémoire centrale. Par conséquent, l'ensemble des règles d'association n'a pas besoin d'être stocké. Ceci réduit grandement la consommation de mémoire et rend aussi l'algorithme plus rapide. De plus, chaque règle séquentielle peut être sauvegardée sur le disque immédiatement après avoir été trouvée, et si l'un désire conserver aussi les règles d'association, ces dernières peuvent aussi être écrites sur le disque en même temps. Ces stratégies d'implémentation réduisent grandement la consommation en mémoire de l'algorithme.

5.4.4 Analyse de la complexité de l'algorithme

Cette section présente une brève analyse de la complexité en temps d'exécution de l'algorithme. Premièrement, le temps pour convertir une base de séquences en base de transactions (étape 1 de l'algorithme) est linéaire par rapport au nombre de séquences et à leur taille.

La complexité en temps de l'étape 2 est plus difficile à établir. Elle dépend de l'algorithme utilisé pour la recherche de règles d'association. Comme expliqué précédemment, la recherche de règles d'association se fait en deux étapes : la recherche d'itemsets fré-

quents et la génération de règles. La première étape est la plus coûteuse [2]. Il est donc adéquat d'ignorer la deuxième étape pour estimer la complexité temporelle. Si l'algorithme Apriori est utilisé pour la recherche d'itemsets fréquents, la complexité temporelle est de $\Theta(d^2 \times n)$ où d est le nombre d'items différents et n est le nombre de transactions dans la base de transactions (voir [86] pour une preuve). Dans l'implémentation réalisée pour cette thèse, la variation Apriori-Tid a été utilisée et performe légèrement mieux pour certaines bases de transactions [2] et à une complexité temporelle similaire [86] .

L'étape 3 de l'algorithme vérifie si chaque règle candidate est incluse dans un ensemble de séquences qui contiennent son antécédent (comme expliqué). Dans le meilleur et dans le pire cas, il y a respectivement $|D| \times \text{minSup}$ séquences et $|D|$ séquences à vérifier pour chaque règle. Vérifier si une règle est contenue dans une séquence est fait en temps linéaire. Par conséquent, la complexité temporelle de l'étape 3 est linéaire par rapport au nombre de séquences qui contiennent l'antécédent de chaque règle, la taille de chaque séquence et le nombre de règles.

5.4.5 Extensions de l'algorithme

Une propriété intéressante de l'algorithme CMRules est qu'il est facile à étendre pour lui donner de nouvelles capacités. Deux types d'extensions sont facilement réalisables.

Extension par le choix d'un autre algorithme de recherche de règles d'association

Tout d'abord, il est possible de donner des capacités supplémentaires à CMRules en choisissant un algorithme d'extraction de règles d'association possédant cette capacité. Par exemple, si quelqu'un implémente l'algorithme CMRules en utilisant un algorithme incrémental tel que celui de Cheung et al. [30], le résultat serait un algorithme incrémental pour découvrir les règles séquentielles. Une deuxième extension possible est d'utiliser un algorithme parallèle comme celui de Agrawal et Shafer [3].

Une troisième extension est d'ajouter des contraintes sur la taille des règles à découvrir.

Cette extension a été intégrée dans l'implémentation de CMRules réalisée pour cette thèse. Quatre types de contraintes ont été incluses : (1) *minLeft*, la taille minimale de l'antécédent d'une règle, (2) *maxLeft*, la taille maximale de l'antécédent d'une règle, (3) *minRight*, la taille minimale du conséquent d'une règle et (4) *maxRight*, la taille maximale du conséquent d'une règle. Pour expliquer cette modification, il faut se rappeler que la génération des règles d'association se fait en deux étapes (cf. section 5.4.2) : la découverte d'itemsets fréquents, puis la sélection de paires d'itemsets fréquents X et Y tel que $X \subseteq Y$, pour générer des règles de la forme $X \rightarrow Y \setminus X$ ou bien $Y \setminus X \rightarrow X$ [2].

Le premier changement est apporté à la recherche d'itemset fréquents. Il est inutile de découvrir des itemsets de taille supérieure à $maxLeft + maxRight$ pour générer toutes les règles respectant ces contraintes. Pour un algorithme comme AprioriTid qui découvre les itemsets par taille croissante de manière récursive, il suffit d'ajouter une condition d'arrêt de la récursion lorsque la taille $maxLeft + maxRight$ est atteinte. Le deuxième changement est appliqué à l'algorithme pour générer les règles d'association à partir des itemsets fréquents. Puisque dans l'algorithme d'Agrawal [2], les règles sont construites récursivement en agrandissant/réduisant la partie gauche ou droite des règles, il est facile d'incorporer les quatre contraintes pour éviter d'agrandir/réduire des règles dont le résultat ne satisfera pas les contraintes. Une dernière modification à faire est d'ajouter une vérification qu'une règle respecte les quatre contraintes avant de calculer sa fréquence séquentielle et sa confiance séquentielle.

Extension par la modification du calcul de la fréquence

Hormis ce type d'extensions, plusieurs autres sont possibles en modifiant le code pour vérifier si une règle est incluse dans une séquence (l'étape 3 de l'algorithme). Par exemple, il est possible de modifier la définition de $sup(X \blacksquare Y)$ pour que la partie gauche et la partie droite d'une règle doivent apparaître à l'intérieur d'un laps de temps, comme cela est proposé dans d'autres travaux comme ceux de Mannila [107], Das et al. [36], Harms [83].

5.5 Algorithme CMDeogun pour la recherche de règles séquentielles communes à plusieurs séquences

Ce chapitre propose un deuxième algorithme nommé CMDeogun [69] pour extraire des règles séquentielles communes à plusieurs séquences. Cet algorithme est une adaptation de l'algorithme de Deogun et Jiang [37]. Ce dernier est un algorithme pour rechercher des règles apparaissant fréquemment à l'intérieur d'une fenêtre temporelle glissante dans une seule séquence.

L'algorithme de Deogun et al. a été adapté de deux façons. Premièrement, pour qu'il découvre des règles communes à plusieurs séquences d'une base de séquences, chaque séquence est considérée comme une occurrence de la fenêtre temporelle dans l'algorithme original. Le deuxième point qui a demandé des modifications est que l'algorithme de Deogun et al. extrait seulement un sous-ensemble des règles séquentielles et qu'il n'y a aucune garantie que cet ensemble permet de reconstituer toutes les règles [37]. Pour éviter cette perte d'information, l'algorithme a été adapté pour extraire toutes les règles. Seul l'algorithme adapté est décrit dans ce chapitre. Pour une description de l'algorithme original, le lecteur pourra se référer à l'article de Deogun et Jiang [37].

5.5.1 Notions préliminaires

Avant de présenter l'algorithme, quelques notions préliminaires sont nécessaires. La *taille d'une règle séquentielle* $r : X \Rightarrow Y$ où les itemsets X et Y contiennent respectivement f et g items, est notée $f \star g$. L'ensemble des règles séquentielles valides de taille $f \star g$ est noté $R_{f \star g}$.

L'algorithme de Deogun et Jiang [37] utilise deux processus pour générer des règles candidates.

L'*expansion par la gauche* consiste à prendre deux règles de taille $f \star g$, $X \Rightarrow Y$ et $Z \Rightarrow Y$ où X et Z sont des itemsets partageant exactement $f - 1$ items, pour générer une règle $X \cup Z \Rightarrow Y$ de taille $(f + 1) \star g$.

L'*expansion par la droite* consiste à prendre deux règles de taille $f \star g$, $Y \Rightarrow X$ et $Y \Rightarrow Z$ où X et Z sont des itemsets partageant exactement $g - 1$ items, pour générer une règle $Y \Rightarrow X \cup Z$ de taille $f \star (g + 1)$.

Deux propriétés peuvent être démontrées pour le processus d'expansion par la gauche :

Proposition PG1 : Une règle $r_3 : X \cup Z \Rightarrow Y$ résultante de l'expansion par la gauche de règles $r_1 : X \Rightarrow Y$ et $r_2 : Z \Rightarrow Y$ a une fréquence séquentielle inférieure ou égale à celle de r_1 et celle de r_2 .

Preuve : La fréquence séquentielle de r_1 , r_2 et r_3 est respectivement $\sup(X \blacksquare Y)/|D|$, $\sup(Z \blacksquare Y)/|D|$ et $\sup(X \cup Z \blacksquare Y)/|D|$. Puisque $\sup(X \cup Z \blacksquare Y) \leq \sup(X \blacksquare Y)$ et $\sup(X \cup Z \blacksquare Y) \leq \sup(Z \blacksquare Y)$, la propriété est démontrée.

Proposition PG2 : Une règle $r_3 : X \cup Z \Rightarrow Y$ résultante de l'expansion par la gauche de règles $r_1 : X \Rightarrow Y$ et $r_2 : Z \Rightarrow Y$ a une confiance séquentielle supérieure, inférieure ou égale à celle de r_1 et celle de r_2 .

Preuve : La confiance séquentielle de r_1 , r_2 et r_3 est respectivement $\sup(X \blacksquare Y)/\sup(X)$, $\sup(Z \blacksquare Y)/\sup(Z)$ et $\sup(X \cup Z \blacksquare Y)/\sup(X \cup Z)$. Pour les numérateurs, il est clair que $\sup(X \cup Z \blacksquare Y) \leq \sup(X \blacksquare Y)$ et $\sup(X \cup Z \blacksquare Y) \leq \sup(Z \blacksquare Y)$. Pour les dénominateurs, on a les relations $\sup(X \cup Z) \leq \sup(X)$ et $\sup(X \cup Z) \leq \sup(Z)$. Puisque le dénominateur et le numérateur de la confiance pour r_3 ne peuvent que diminuer par rapport à ceux de r_1 et r_2 , la confiance séquentielle de r_3 peut-être supérieure, inférieure ou égale à celle de r_1 et celle de r_2 .

Deux propriétés peuvent être démontrée pour le processus d'expansion par la droite :

Proposition PD1 : Une règle $r_3 : Y \Rightarrow X \cup Z$ résultante de l'expansion par la droite de règles $r_1 : Y \Rightarrow X$ et $r_2 : Y \Rightarrow Z$ a une fréquence séquentielle inférieure ou égale à celle de r_1 et celle de r_2 .

Preuve : La fréquence séquentielle de r_1 , r_2 et r_3 est respectivement $\sup(Y \blacksquare X)/|D|$, $\sup(Y \blacksquare Z)/|D|$ et $\sup(Y \blacksquare X \cup Z)/|D|$. Puisque $\sup(Y \blacksquare X \cup Z) \leq \sup(Y \blacksquare X)$ et $\sup(Y \blacksquare X \cup Z) \leq \sup(Y \blacksquare Z)$, la propriété est démontrée.

Proposition PD2 : Une règle $r_3 : Y \Rightarrow X \cup Z$ résultante de l'expansion par la droite

de règles $r_1 : Y \Rightarrow X$ et $r_2 : Y \Rightarrow Z$ a une confiance séquentielle inférieure ou égale à celle de r_1 et celle de r_2 . *Preuve* : La confiance séquentielle de r_1 , r_2 et r_3 est respectivement $\text{sup}(Y \blacksquare X)/\text{sup}(Y)$, $\text{sup}(Y \blacksquare Z)/\text{sup}(Y)$ et $\text{sup}(Y \blacksquare X \cup Z)/\text{sup}(Y)$. Puisque $\text{sup}(Y \blacksquare X \cup Z) \leq \text{sup}(Y \blacksquare X)/\text{sup}(Y)$ et $\text{sup}(Y \blacksquare X \cup Z) \leq \text{sup}(Y \blacksquare Z)/\text{sup}(Y)$, la propriété est démontrée.

5.5.2 Description de l'algorithme CMDeogun

L'algorithme est présenté à la figure 5.5. L'algorithme prend en paramètres une base de séquences et les seuils minSeqSup et minSeqConf . Le principe général de l'algorithme est de trouver tout d'abord les règles de taille $1 \star 1$, puis d'utiliser des expansions par la gauche et des expansions par la droite pour générer progressivement des règles de tailles supérieures et trouver toutes les règles valides.

L'algorithme utilise les propriétés $PG1$, $PG2$, $PD1$ et $PD2$ pour éviter de faire des expansions qui sont garanties de ne pas permettre de trouver des règles valides. Les expansions ne permettant pas d'atteindre de nouvelles règles valides à partir d'une règle r sont (1) l'expansion par la gauche lorsque $\text{seqSup}(r) < \text{minSeqSup}$ (par la propriété $PG1$), (2) l'expansion par la droite lorsque $\text{seqSup}(r) < \text{minSeqSup}$ (par la propriété $PD1$) et (3) l'expansion par la droite lorsque $\text{seqConf}(r) < \text{minSeqConf}$ (par la propriété $PD2$). Toutefois, il est impossible de prédire si l'expansion par la gauche d'une règle r tel que $\text{seqConf}(r) < \text{minSeqConf}$ permettra d'atteindre des règles valides (par la propriété $PG2$). Pour cette raison, chaque règle r telle que $\text{seqSup}(r) > \text{minSeqSup}$ sera étendue par la gauche et par la droite même si $\text{seqConf}(r) < \text{minSeqConf}$. En d'autres mots, ce ne sont pas seulement les règles valides qui seront étendues. Dans l'algorithme, l'ensemble des règles séquentielles d'une taille $f \star g$ avec une fréquence séquentielle supérieure ou égale à minSeqSup est noté $C_{f \star g}$, alors que l'ensemble des règles valides est nommé $AllRules$.

Pour trouver les règles de taille $1 \star 1$, l'algorithme procède tout d'abord en parcourant la base de séquences pour noter l'ensemble des séquences, appelé $sidset$, contenant chaque

item (instruction 3). Ensuite, l'algorithme tente de générer des règles de taille $1 \star 1$. Cela est fait en bouclant avec chaque paire d'items fréquents x, y pour voir si $x \neq y$ et $|x.sidset \cap y.sidset| \geq minSeqSup$ (pour s'assurer qu'une règle résultante peut avoir la fréquence minimale). Si ce test est passé alors des règles $r_{xy} := x \Rightarrow y$ et $r_{yx} := y \Rightarrow x$ sont créées (instruction 4.a) puis les séquences $|x.sidset \cap y.sidset|$ sont parcourues pour calculer $sup(x \blacksquare y)$ et $sup(y \blacksquare x)$, les valeurs nécessaires pour calculer la confiance et la fréquence séquentielle. Ce calcul est fait exactement comme l'étape 3 de l'algorithme CMRules. Après, si $seqsup(r_{xy}) > minseqSup$, la règle est ajoutée à $C_{1 \star 1}$, et si $seqConf(r_{xy}) \geq minSeqConf$, la règle est ajoutée à *AllRules* (instruction 4.b). La même chose est répétée pour la règle r_{yx} (instruction 4.c). Finalement, une fois la boucle terminée, la procédure d'expansion par la gauche et d'expansion par la droite est appelée avec les règles de $C_{1 \star 1}$ (instructions 5 et 6).

Les procédures d'expansion par la gauche et d'expansion par la droite sont présentées respectivement aux figures 5.6 et 5.7. Elle prennent en paramètres la base de séquences D , les seuils $minSeqSup$, $minSeqConf$, un ensemble de règles à étendre $C_{k \star m}$ et *AllRules*.

La procédure d'expansion par la gauche est la suivante. Une boucle est faite avec chaque paire de règles $r_x := X \Rightarrow Y, r_y := Z \Rightarrow Y$ tel que $r_x, r_y \in C_{k \star m}$, et X et Z possèdent exactement $k - 1$ items en commun (instruction 2). Cette dernière condition permet de s'assurer que l'expansion donnera une règle de taille $(k + 1) \star m$. Puis, la règle $r_{xy} := X \cup Z \Rightarrow Y$ est créée (instruction 2.a). Ensuite, chaque séquence de $(X \cup Z).sidset$ est parcourue pour calculer $sup(X \cup Z \blacksquare Y)$ nécessaire au calcul de $seqSup(r_{xy})$ et $seqConf(r_{xy})$ (instructions 2.a et 2.c). Ce calcul est fait exactement comme l'étape 3 de l'algorithme CMRules. Après, si $seqsup(r_{xy}) > minseqSup$, la règle est ajoutée à $C_{(k+1) \star m}$, et si $seqConf(r_{xy}) \geq minSeqConf$, la règle est ajoutée à *AllRules* (instruction 2.d). Finalement, une fois la boucle terminée, la procédure d'expansion par la gauche est appelée avec les règles de $C_{(k+1) \star m}$ (instructions 3).

La procédure d'expansion par la droite suit exactement le même principe, excepté qu'une expansion par la gauche et une expansion par la droite sont réalisées à la fin de la

procédure (instruction 3 et 4 de *expansionDroite*) plutôt qu'une seule expansion par la gauche (instruction 3 de *expansionGauche*). Cette différence permet d'éviter qu'à partir d'un ensemble de règle $C_{k \star m}$, une expansion par la gauche suivi d'une expansion par la droite résultera en l'exploration de règles de taille $(k + 1) \star (m + 1)$, alors qu'une expansion par la droite suivie d'une expansion par la gauche explorera aussi les règles de taille $(k + 1) \star (m + 1)$, et que des règles redondantes soient trouvées. Il a été choisi de favoriser l'expansion par la droite plutôt que l'expansion par la gauche, parce que les propriétés *PD1* et *PD2* sont plus avantageuses que les propriétés *PG1* et *PG2* (*PG2* ne permet pas de garantir que l'expansion par la droite d'une règle r tel que $seqConf(r) < minSeqConf$ permettra d'atteindre des règles valides).

Puisque l'algorithme procède par niveau en augmentant graduellement la taille des règles en suivant les quatre propriétés, il est garanti que l'algorithme trouvera toutes les règles. Au niveau de l'implémentation, il est bénéfique d'ordonner les items dans les itemsets selon un ordre lexicographique. Cela permet de vérifier rapidement que deux règles peuvent être combinées pour faire une expansion en vérifiant si les $k - 1$ premiers éléments des itemsets X et Z à comparer sont les mêmes. Si les itemsets ne sont pas ordonnés, cette comparaison est plus coûteuse.

5.5.3 Extensions de l'algorithme

Tout comme l'algorithme CMRules, l'algorithme CMDeogun peut être étendu de plusieurs façons. Deux types d'extensions sont ici abordées.

Extension par l'ajout de contraintes dans les expansions

Une première façon d'étendre l'algorithme CMDeogun est d'ajouter des contraintes dans le processus d'expansion des règles. Un exemple est l'ajout de contraintes sur la taille des règles à découvrir. Les contraintes *minLeft*, *maxLeft*, *minRight* et *maxRight* présentées à la section 5.4.5 pour CMRules peuvent être facilement ajoutées à CMDeogun. Pour ajouter les contraintes *maxLeft* et *maxRight*, CMDeogun doit être modifié pour ne pas

algoCMDeogun($D, minSeqSup, minSeqConf$)

1. $AllRules := \emptyset$.
2. $C_{1*1} := \emptyset$.
3. Parcourir D pour noter l'ensemble des séquences (sidsets) contenant chaque item.
4. POUR CHAQUE paire d'items fréquents x, y tel que $x \neq y$ et $|x.sidset \cap y.sidset| \geq minSeqSup$:
 - (a) Soit $r_{xy} := x \Rightarrow y$ et $r_{yx} := y \Rightarrow x$. Parcourir chaque séquence $t \in x.sidset \cap y.sidset$ de D , et noter le sidset de $x \blacksquare y$ et $y \blacksquare x$ afin de calculer la fréquence séquentielle et la confiance séquentielle de r_{xy} et r_{yx} .
 - (b) SI $seqSup(r_{xy}) \geq minSeqSup$ ALORS
 - i. $C_{1*1} = C_{1*1} \cup \{r_{xy}\}$
 - ii. SI $seqConf(r_{xy}) \geq minSeqConf$ ALORS $AllRules := AllRules \cup \{r_{xy}\}$.
 - (c) SI $seqSup(r_{yx}) \geq minSeqSup$ ALORS
 - i. $C_{1*1} = C_{1*1} \cup \{r_{yx}\}$
 - ii. SI $seqConf(r_{yx}) \geq minSeqConf$ ALORS $AllRules := AllRules \cup \{r_{yx}\}$.
5. $expansionGauche(D, minSeqSup, minSeqConf, C_{1*1}, AllRules)$
6. $expansionDroite(D, minSeqSup, minSeqConf, C_{1*1}, AllRules)$

Figure 5.5 Algorithme CMDeogun pour l'extraction de règles séquentielles

$expansionGauche(D, minSeqSup, minSeqConf, C_{k*m}, AllRules)$

1. $C_{(k+1)*m} := \emptyset$.
2. POUR chaque paire de règles $r_x := X \Rightarrow Y, r_y := Z \Rightarrow Y$ tel que $r_x, r_y \in C_{k*m}$, et X et Z possèdent exactement $k - 1$ items en commun.
 - (a) $r_{xy} := X \cup Z \Rightarrow Y$.
 - (b) $(X \cup Z).sidset := X.sidset \cap Z.sidset$
 - (c) Parcourir chaque séquence $t \in (X \cup Z).sidset$ de D pour calculer $sup(X \cup Z \blacksquare Y)$ nécessaire au calcul de $seqSup(r_{xy})$ et $seqConf(r_{xy})$.
 - (d) SI $seqSup(r_{xy}) \geq minSeqSup$ ALORS
 - i. $C_{(k+1)*m} = C_{(k+1)*m} \cup \{r_{xy}\}$
 - ii. SI $seqConf(r_{xy}) \geq minSeqConf$ ALORS $AllRules := AllRules \cup \{r_{xy}\}$.
3. $expansionGauche(D, minSeqSup, minSeqConf, C_{k*(m+1)}, AllRules)$

Figure 5.6 Algorithme $expansionGauche$

$\text{expansionDroite}(D, \text{minSeqSup}, \text{minSeqConf}, C_{k\star m}, \text{AllRules})$

1. $C_{k\star(m+1)} := \emptyset$.
2. POUR chaque paire de règles $r_x := Y \Rightarrow X, r_y := Y \Rightarrow Z$ tel que $r_x, r_y \in C_{k\star m}$, et X et Z possèdent exactement $k - 1$ items en commun.
 - (a) $r_{xy} := Y \Rightarrow X \cup Z$.
 - (b) $(X \cup Z).\text{sidset} := X.\text{sidset} \cap Z.\text{sidset}$
 - (c) Parcourir chaque séquence $t \in X.\text{sidset}$ de D pour calculer $\text{sup}(X \cup Z \blacksquare Y)$ nécessaire au calcul de $\text{seqSup}(r_{xy})$ et $\text{seqConf}(r_{xy})$.
 - (d) SI $\text{seqSup}(r_{xy}) \geq \text{minSeqSup}$ ALORS
 - i. $C_{k\star(m+1)} = C_{k\star(m+1)} \cup \{r_{xy}\}$
 - ii. SI $\text{seqConf}(r_{xy}) \geq \text{minSeqConf}$ ALORS $\text{AllRules} := \text{AllRules} \cup \{r_{xy}\}$.
3. $\text{expansionGauche}(D, \text{minSeqSup}, \text{minSeqConf}, C_{k\star(m+1)}, \text{AllRules})$
4. $\text{expansionDroite}(D, \text{minSeqSup}, \text{minSeqConf}, C_{k\star(m+1)}, \text{AllRules})$

Figure 5.7 Algorithme *expansionDroite*

faire d'expansion vers la gauche/droite avec les règles dont la partie gauche/droite est égale à *maxLeft*/*maxRight*. Pour ajouter les contraintes *minLeft* et *minRight*, CMDeogun doit être modifié pour vérifier que chaque règle valide respecte ces contraintes avant de l'ajouter à *AllRules*.

Extension par la modification du calcul de la fréquence

Comme pour CMRules (cf. section 5.4.5), il est possible d'étendre l'algorithme CMDeogun en modifiant le code pour vérifier si une règle est incluse dans une séquence. Comme mentionné, il est possible par exemple de modifier la définition de $\text{sup}(X \blacksquare Y)$ pour que la partie gauche et la partie droite d'une règle doivent apparaître à l'intérieur d'un laps de temps, comme cela est proposé dans d'autres travaux comme ceux de Mannila [107], Das et al. [36], Harms [83].

5.6 Évaluation comparative des performances des algorithmes

Les deux algorithmes ont été implémentés en Java. Cette section décrit cinq expériences pour comparer la performance des algorithmes. Des bases de séquences publiques prove-

	Kosarak-1	Kosarak-2	BMS-WebView1	Toxin-Snake
Nb. séquences	70000	41582	59601	163
Nb. items	21144	25257	497	20
Nb. items/itemset	1	3	1	1
Nb. items/séquences	7.97 $\sigma = 21.14$	26.51 $\sigma = 28.25$	2.51 $\sigma = 4.85$	17.84 $\sigma = 1.09$

Tableau 5.1 Caractéristiques des jeux de données

nant d'applications réelles ont été utilisées pour cette comparaison. Elles ont été choisies parce qu'elles sont des bases de séquences connues, avec des caractéristiques variées et qu'elles sont accessibles librement. Elles ont été utilisées en fouille de données pour comparer la performance de divers algorithmes comme des algorithmes de recherche de motifs séquentiels et des algorithmes de recherche d'itemsets fréquents [147, 162].

5.6.1 Les bases de séquences utilisées

Le tableau 5.1 illustre les caractéristiques des bases de séquences utilisées pour les expériences. Les bases de séquences se nomment Kosarak1, Kosarak2, BMS-WebView1 et Toxin-Snake. Pour chaque base de séquences, les quatre lignes de la table indiquent le nombre de séquences, le nombre total d'items, le nombre d'items par itemset et le nombre moyen d'items par séquence.

Les deux premières bases de séquences, Kosarak1 et Kosarak2, ont été dérivées de la base de séquences Kosarak disponible à l'adresse Web <http://fimi.cs.helsinki.fi/data/>. Kosarak est une base de séquences contenant 990000 séquences de navigation d'utilisateurs sur un portail de nouvelles hongrois. Kosarak-1 est une version réduite de Kosarak contenant seulement les 70000 premières séquences de Kosarak. Les 70000 premières séquences ont été choisies pour accélérer la réalisation de l'expérience. Ces séquences contiennent un item par itemset et en moyenne 7.97 itemsets par séquence ($\sigma \equiv 21.145$) parmi 21144 items. Dans Kosarak-1, chaque item apparaissant dans une séquence s'y trouve en moyenne une seule fois ($\sigma = 0.003$).

Kosarak-2 est une autre variation de Kosarak. Puisque la base de séquences Kosarak contient plusieurs séquences courtes, il a été décidé de ne pas considérer les séquences contenant moins de 10 items. Ceci a résulté en 41582 séquences longues contenant en moyenne 26.51 items ($\sigma = 28.25$) pour un total de 25257 items différents. De plus, puisque les séquences de Kosarak, ne contiennent qu'un seul item par itemset, il a été décidé de transformer les séquences pour grouper les items en groupes de trois pour former des itemsets de taille trois. Le nombre de trois a été choisi, car il est petit et que par conséquent cela permet aux séquences de contenir plusieurs itemsets. Pour réaliser le groupement, lorsque le nombre d'items d'une séquence n'était pas un multiple de trois, les derniers items ont été omis pour que le nombre d'items deviennent un multiple de trois. Par exemple, une séquence contenant 23 items a été transformée en sept itemsets de trois items et les deux derniers items ont été omis. Dans Kosarak-2, chaque item apparaissant dans une séquence s'y trouve en moyenne une seule fois ($\sigma = 0.003$).

La troisième base de séquences est BMS-WebView1, une base de séquences employée pour la compétition KDD-2000. BMS-WebView1 peut être téléchargée à l'adresse Web : <http://www.ecn.purdue.edu/KDDCUP/>. Elle contient 59601 séquences de navigation d'un commerce en ligne. Les séquences ont en moyenne 2.51 items ($\sigma = 4.85$) parmi 497 items, et chaque item n'apparaît jamais plus qu'une fois par séquence. Chaque item représente la visite d'une page Web d'un produit. BMS-WebView1 se distingue de Kosarak-1 et Kosarak-2 principalement par la courte taille des séquences et le nombre d'items qui est beaucoup plus restreint.

Le quatrième base de séquences est Toxin-Snake [92]. Cette base de séquences du domaine de la biologie contient 192 séquences de protéines. Pour l'expérience, seules les séquences de taille supérieure à 50 ont été employées, soit 163 séquences avec une taille moyenne de 60.61 items ($\sigma = 0.598$). Conserver seulement ces séquences a été fait pour uniformiser les caractéristiques de la base de séquences, les autres séquences étant très courtes (à peine quelques items). La particularité de Toxin-Snake est qu'elle est très dense : chaque item est présent dans presque toutes les séquences (en moyenne, il y a 17.84 items différents par séquence) et les items apparaissant dans une séquence y sont présents en moyenne

3.39 fois ($\sigma = 2.24$).

Les prochaines sous-sections présentent les résultats de chaque expérience puis une analyse des résultats.

5.6.2 Première expérience : Kosarak-1

Pour la première expérience, les deux algorithmes ont été appliqués sur Kosarak-1 avec $minSeqConf = 0.3$ et $minSeqSup = 0.02, 0.019, \dots, 0.004$. La figure 5.8(A) montre le temps d'exécution de chaque algorithme. Lorsque $minSeqSup$ est élevé, les deux algorithmes ont eu une performance comparable, mais lorsque $minSeqSup$ diminue, CMRules est devenu considérablement plus rapide. Par exemple, lorsque $minSeqSup = 0.02$, les deux algorithmes avaient une performance similaire, alors que pour $minSeqSup = 0.04$, l'algorithme CMRules était environ cinq fois plus rapide.

Pendant l'expérience, le nombre de règles d'association et le nombre de règles séquentielles découvertes ont aussi été mesurés. La figure 5.8(B) illustre ces nombres en fonction de $minSeqSup$. Le nombre de règles séquentielles a varié entre 11 à 20 % de l'ensemble des règles d'association. Ceci montre que le pourcentage de règles séquentielles peut être relativement élevé dans des bases de séquences provenant de données réelles. Par exemple, pour $minSeqSup = 0.02$, il y avait 312 règles d'association et 64 règles séquentielles (20.51 %), et pour $minSeqSup = 0.004$, il y avait 10011 règles d'association et 1169 règles séquentielles (11.68 %). Il a aussi été observé que le temps d'exécution de l'algorithme CMRules semble croître proportionnellement au nombre de règles d'association. Au niveau de la taille des règles, à $minSeqSup = 0.003$, le nombre d'items moyen par règle séquentielle était 3.72 ($\sigma = 1.01$, maximum : 6).

5.6.3 Deuxième expérience : Kosarak-2

Les deux algorithmes ont été appliqués sur la base de séquences Kosarak-2 avec $minSeqConf = 0.3$ et $minSeqSup = 0.15, 0.145 \dots 0.065$. La figure 5.9(A) illustre les temps d'exécution en fonction de $minSeqSup$. Comme pour la première expérience,

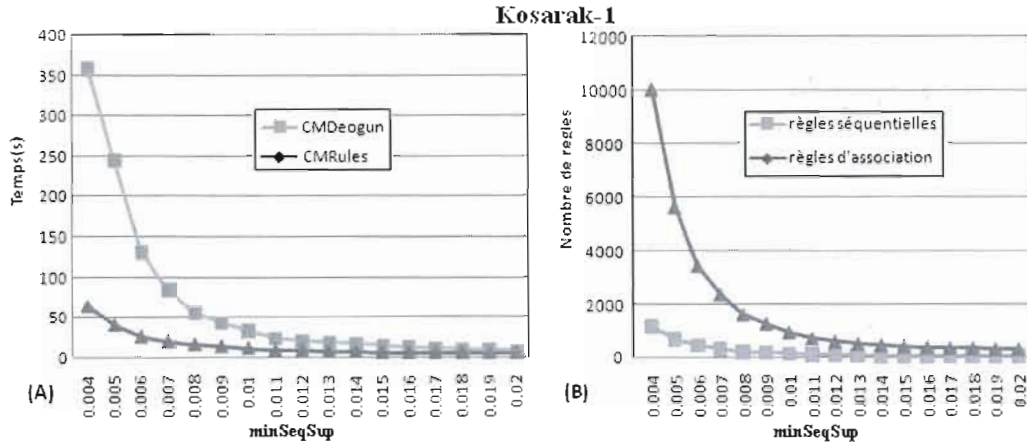


Figure 5.8 Résultats de l'expérience avec Kosarak-1

l'algorithme CMRules a montré une meilleure performance par rapport à CMDeogun quand $minSeqSup$ a diminué. Aussi, le temps d'exécution de CMRules semble croître proportionnellement au nombre de règles d'association de façon plus visible que dans la première expérience. La figure 5.9(B) illustre le nombre de règles d'association et le nombre de règles séquentielles. Le pourcentage de règles séquentielles a varié entre 9 % et 14 %. Au niveau de la taille des règles, à $minSeqSup = 0.065$, le nombre d'items moyen par règle séquentielle était 3.64 ($\sigma = 1$, maximum : 7).

5.6.4 Troisième expérience : BMS-WebView1

Pour la troisième expérience, les deux algorithmes ont été appliqués sur la base de séquences BMS-WebView1 avec $minSeqConf = 0.2$ et $minSeqSup = 0.0007, 0.00075, \dots, 0.00135$. La figure 5.10(A) illustre les temps d'exécution en fonction de $minSeqSup$. CMDeogun a obtenu une meilleure performance que CMRules lorsque $minSeqSup$ était inférieur à 0.00085. La figure 5.10(B) montre le nombre de règles d'association et de règles séquentielles. Le pourcentage de règles séquentielles a varié entre 7 % et 29 %. En comparant la partie (A) et (B) des graphiques, il est possible de voir clairement que le temps d'exécution de CMRules a varié proportionnellement au nombre de règles d'association. Au niveau de la taille des règles, à $minSeqSup = 0.00135$, le nombre

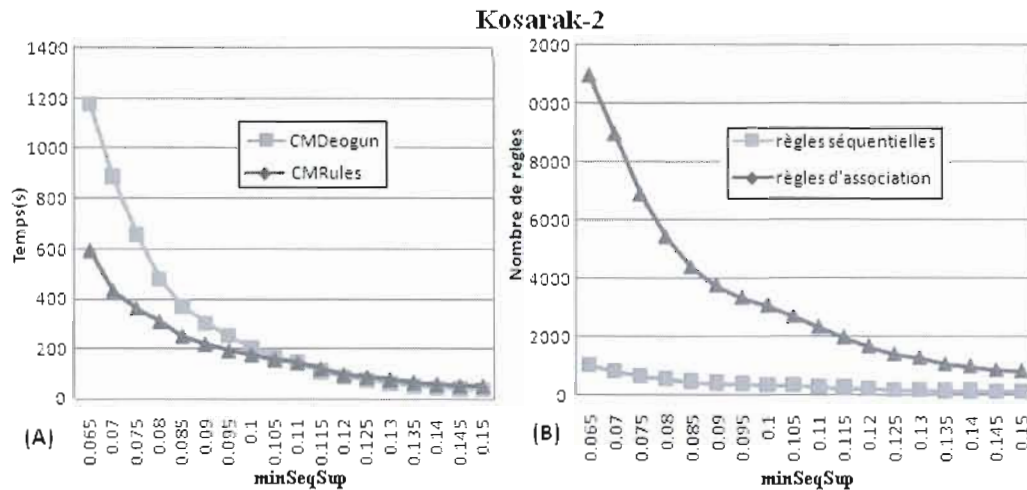


Figure 5.9 Résultats de l'expérience avec Kosarak-2

d'items moyen par règle séquentielle était 5.13 ($\sigma = 1.57$, maximum : 11).

5.6.5 Quatrième expérience : Toxin-Snake

Les deux algorithmes ont été appliqués sur la base de séquences Toxin-Snake avec $minSeqConf = 0.2$ et $minSeqSup = 0.985, 0.98...0.93$. La figure 5.11(A) illustre les temps d'exécution de chaque algorithme en fonction de $minSeqSup$. La figure 5.11(B) illustre le nombre de règles d'association et de règles séquentielles. Le pourcentage de règles séquentielles a varié entre 2 % et 11 %. Le nombre d'items moyen par règle séquentielle était 5.06 ($\sigma = 1.29$, maximum : 10).

Lors de cette expérience, CMDeogun a montré une performance supérieure à celle de CMRules en tout temps et davantage lorsque $minSeqSup$ a diminué. L'écart de performance entre CMRules et CMDeogun est plus apparent que pour l'expérience avec BMS-WebView1. Lorsque $minSeqSup = 0.93$, CMDeogun a été 20 fois plus rapide que CMRules.

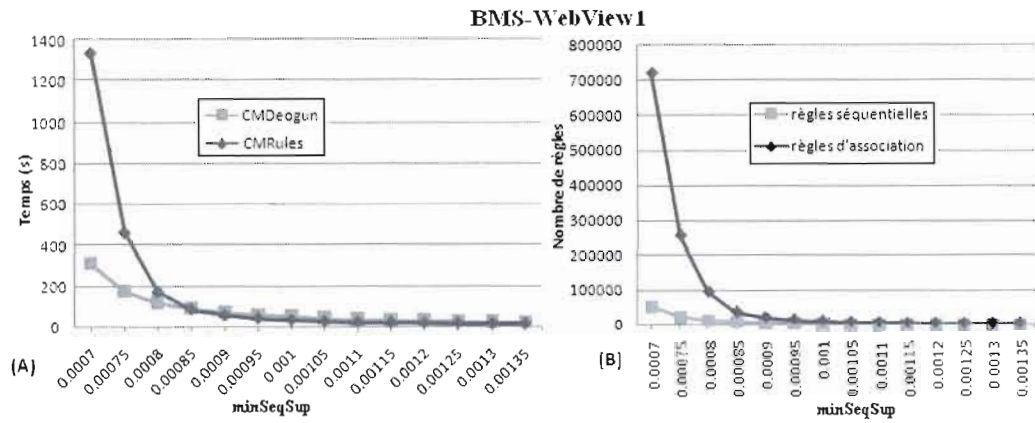


Figure 5.10 Résultats de l'expérience avec BMS-WebView1

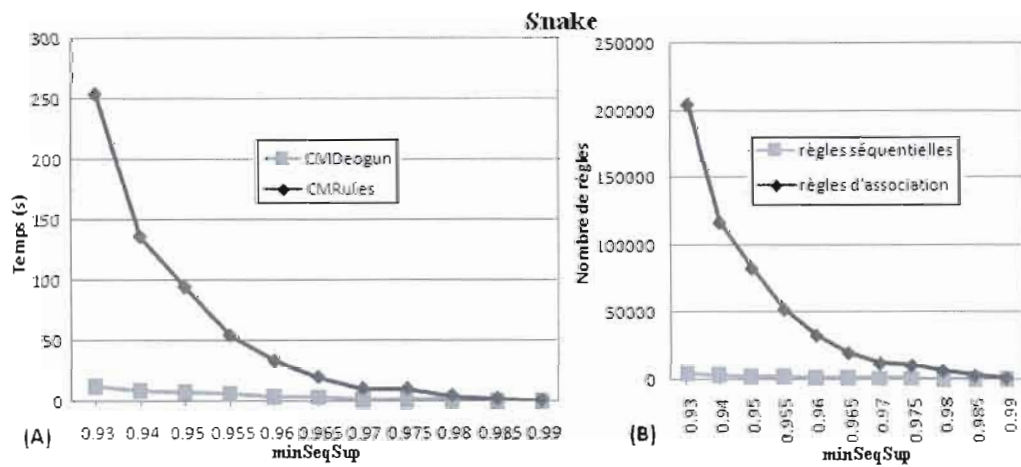


Figure 5.11 Résultats de l'expérience avec Toxin-Snake

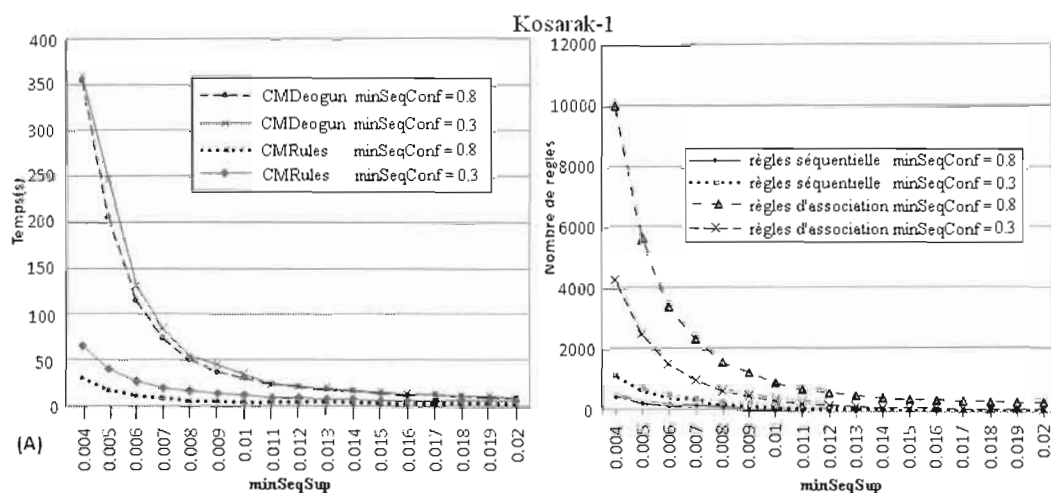


Figure 5.12 Influence de *minSeqConf* avec Kosarak-1

5.6.6 Cinquième expérience : Kosarak-1, *minSeqConf* = 0.8

Les quatre premières expériences ont consisté à appliquer CMRules et CMDeogun avec une valeur constante pour *minSeqConf* tout en faisant varier *minSeqSup*. Cette expérience étudie l'influence de la *minSeqConf*.

Les deux algorithmes ont été appliqués sur Kosarak-1 avec *minSeqConf* = 0.8 et *minSeqSup* = 0.02, 0.019, ..., 0.004. La figure 5.12(A) compare le temps d'exécution de chaque algorithme pour *minSeqConf* = 0.8 (cette expérience) et *minSeqConf* = 0.3 (les résultats de la première expérience). La figure 5.12(B) illustre le nombre de règles séquentielles et de règles d'association pour *minSeqConf* = 0.8 et *minSeqConf* = 0.3.

Pour *minSeqConf* = 0.8, le nombre de règles d'association a baissé de 58 % à 71 % (en moyenne : 62.01 %) par rapport à *minSeqConf* = 0.3, et le nombre de règle séquentielle a diminué de 55 % à 67 % (en moyenne : 62 %). La performance en temps d'exécution de CMRules a été améliorée de 42 % à 53 % (en moyenne : 48.01 %) et celle de CMDeogun, seulement de 2 % à 13 % (en moyenne : 7.13 %).

5.6.7 Analyse des résultats

CMRules et CMDeogun ont retourné exactement le même nombre de règles dans toutes les expériences, ce qui montre que les implémentations génèrent les résultats attendus. CMrules et CMDeogun ont été chacun plus efficaces pour certaines bases de séquences. CMrules s'est montré plus efficace pour les bases de séquences Kosarak-1 et Kosarak-2, alors que CMDeogun l'a été pour BMS-WebView1 et Toxin-Snake.

Le tableau 5.2 présente une analyse des résultats des quatre premières expériences selon trois critères. La première ligne du tableau indique le pourcentage de règles d'association qui étaient des règles séquentielles. Les deux lignes suivantes indiquent pour la plus petite valeur de *minSeqSup* utilisée pour chaque expérience : (1) le pourcentage des expansions qui ont généré des règles valides dans l'exécution de l'algorithme de CMDeogun, et (2) la fréquence relative moyenne des règles séquentielles trouvées.

La grande efficacité de CMDeogun par rapport à CMrules pour BMS-WebView1 et Toxin-Snake est en grande partie attribuable au fait que les expansions de CMDeogun ont produit une plus grande proportion de règles valides que pour Kosarak-1 et Kosarak-2 (ligne 2 du tableau 5.2). Par exemple, 43.56% des expansions pour Toxin-Snake ont résulté en une règle valide, mais seulement 0.75% pour Kosarak-1. Dans le cas de Toxin-Snake, cela veut dire que presque une règle sur deux générée par une expansion était valide, ce qui a réduit grandement le coût de la découverte des règles valides pour CMDeogun.

Pour CMrules, il a été observé que le temps d'exécution croît proportionnellement avec le nombre de règles d'association. Cela a toutefois été plus visible sur les graphiques de la troisième et quatrième expériences où les courbes de temps d'exécution de CMrules suivent clairement celles du nombre de règles d'association. Une autre observation sur CMrules est que le temps requis pour convertir une base de séquences en une base de transactions (étape 1 de l'algorithme CMrules) a toujours été négligeable par rapport au temps total. Par exemple, ce coût en temps a été inférieur à 300 ms peu importe la

valeur de $minSeqSup$ lors de la première expérience.

La pire performance de CMRules a été pour Toxin-Snake. La principale raison est que chaque item est présent dans presque toutes les séquences et y apparaît généralement plusieurs fois. Ces facteurs et la position des items dans les séquences fait qu'il y a un très grand nombre de règles d'association par rapport au nombre de règles séquentielles. Par exemple, pour $minSeqSup = 0.93$, il y a 204986 règles d'association et seulement 4626 règles séquentielles (2 % des règles d'association). Or, CMRules est désavantagé quand le pourcentage des règles d'association qui sont des règles séquentielles est bas puisque CMRules passera beaucoup de temps à calculer la confiance et fréquence séquentielle pour des règles non valides. Ce calcul est d'autant plus coûteux dans le cas de Toxin-Snake puisque chaque règle d'association est présente dans la presque totalité des séquences (ligne 3 du tableau 5.2). Par conséquent, CMRules doit parcourir la presque totalité des séquences pour calculer $sup(X \blacksquare Y)$ pour chaque règle.

Lors de la cinquième expérience, l'influence de la confiance a été observée avec Kosarak-1. Augmenter $minSeqSup$ de 0.3 à 0.8 a accéléré l'algorithme CMRules en moyenne de 48.01 % et CMDeogun de seulement 7.13 %. La raison de l'amélioration de la performance de CMRules est que le nombre de règles d'association a diminué de façon considérable (en moyenne de 62.01 %) et que pour chacune de ces règles en moins, CMRules n'a pas eu besoin de parcourir la base de séquences pour calculer $sup(X \blacksquare Y)$. Pour CMDeogun, la différence a été moindre puisque $minSeqConf$ est un critère qui permet d'éviter des expansions par la droite grâce à la propriété $PD2$, mais ne permet pas d'éviter des expansions par la gauche à cause de la propriété $PG2$ (cf. section 5.5.2). C'est la raison pourquoi CMDeogun ne peut pas profiter autant du critère $minSeqConf$ que CMRules pour améliorer sa performance.

En conclusion, les deux algorithmes se sont montrés efficaces pour certaines bases de séquences. Pour une application donnée, l'un ou l'autre des deux algorithmes performera mieux en fonction de la nature des séquences à analyser. Les caractéristiques déterminantes de la performance des algorithmes sont la densité des séquences, la proportion

	Exp. 1 Kosarak-1	Exp. 2 Kosarak-2	Exp. 3 BMS-WebView1	Exp. 4 Toxin-Snake
Nb. règles séquentielles / Nb. règles d'association	11 à 20 %	9 à 14 %	7 à 29 %	2 à 11 %
% d'expansions valides (CMDeogun)	0.75 %	1.44 %	11.79 %	43.56 %
Fréquence des règles seq.	0.58 %	10.51 %	0.08 %	60.136 %

Tableau 5.2 Observations pendant les expériences

de règles d'association qui sont des règles séquentielles et la proportion des expansions résultant en des règles valides. Puisque les deux dernières caractéristiques peuvent être difficile à estimer, la meilleure façon de choisir un des deux algorithmes pour une base de séquences est de les tester. La principale faiblesse de CMRules est que son temps d'exécution croît proportionnellement au nombre de règles d'association alors que celle de CMDeogun est que la recherche par expansions génère parfois une grande proportion de règles candidates invalides (des règles ne satisfaisant pas les critères *minSeqSup* et *minSeqConf*), ce qui augmente le coût de la découverte des règles valides. Il a aussi été observé que CMRules peut davantage tirer profit du seuil *minSeqConf* que CMDeogun pour améliorer sa performance.

5.7 Intégration à l'approche d'acquisition automatique

Les algorithmes CMRules et CMDeogun ont été intégrés à l'approche d'acquisition automatique. Cette section décrit en premier l'intégration dans le processus d'extraction de connaissances du domaine, puis une comparaison des performances des deux algorithmes dans CanadarmTutor et finalement l'adaptation des services tutoriels pour exploiter les règles séquentielles découvertes.

5.7.1 Intégration dans le processus d'extraction des connaissances

Dans l'approche d'acquisition automatique, des MD-séquences de deux types sont enregistrées pour un exercice : (1) des MD-séquences d'états et (2) des MD-séquences

d'actions pour passer d'un état à un état adjacent (cf. section 4.6). Pour un exercice cela résulte en une base de MD-séquences d'états (correspondant chacune à une trace d'utilisation), et plusieurs bases de MD-séquences d'actions. Les motifs extraits de ces bases de MD-séquences sont appelés respectivement motifs d'états et motifs d'actions. Ce sont des motifs multidimensionnels (ils sont composés d'un MD-Pattern et d'un motif séquentiel). Pour extraire des règles séquentielles liant des motifs d'actions pouvant apparaître dans différentes bases de MD-séquences d'actions, il a été décidé de construire une *métabase de séquences*, une base de séquences où chaque séquence correspond à une trace d'utilisation et contient des items représentant les motifs d'actions trouvés pour cette trace d'utilisation. Cette base est construite de la façon suivante.

Tout d'abord, une métabase de séquences *MetaBase* est créée avec des séquences vides, appelées *métaséquences*, correspondant chacune à une trace d'utilisation. Ensuite, une boucle est faite pour ajouter chaque motif d'actions dans les métaséquences qui correspondent aux traces d'utilisation où il est présent. Deux défis se posent pour ajouter un motif d'action q dans une métaséquence.

Le premier défi est de savoir quelles traces d'utilisation contiennent q pour déterminer dans quelles métaséquences ajouter q . Pour obtenir cette information, il faut légèrement adapter l'approche d'acquisition automatique présentée au chapitre 4. Tout d'abord, chaque MD-séquence d'actions d'une base de MD-séquences d'actions doit être identifiée par le numéro de la trace d'utilisation correspondante. Ensuite, il faut utiliser un algorithme de recherche de motifs multidimensionnels qui annote chaque motif d'action découvert avec les numéros des MD-séquences d'actions qui le contiennent. Dans l'implémentation de cette thèse SeqDim est employé pour extraire les motifs multidimensionnels. SeqDim procède en deux étapes : (1) trouver les motifs séquentiels fréquents et (2) trouver des MD-Patterns fréquents pour les motifs séquentiels fréquents (cf. section 4.5.1). Dans l'implémentation, l'algorithme d'Hirate et Yamana étendu et Charm sont respectivement utilisés pour faire ces deux tâches (cf. chapitre 4). Il est trivial de modifier l'algorithme d'Hirate et Yamana étendu pour que les motifs séquentiels soient annotés avec les numéros de séquences où ils sont trouvés, et Charm le fait par défaut

pour les MD-Pattern fréquents. Puisque SeqDim procède par la recherche des motifs séquentiels en premier, l'ensemble des numéros de trace d'utilisation contenant un motif multidimensionnel m sera l'ensemble des numéros de trace contenant le MD-Pattern de m . Alternativement, si DimSeq avait été utilisé, l'ensemble des numéros de trace d'utilisation contenant un motif multidimensionnel m serait l'ensemble des numéros de trace associé aux motifs séquentiels de m , tel que trouvé par l'algorithme d'Hirate et Yamana étendu.

Sachant les traces d'utilisation contenant chaque motif d'action, le second défi est de savoir où insérer chaque motif d'action p dans les métaséquences pour que les motifs d'actions y soient ordonnés selon le temps. Cette thèse résout ce problème avec l'approche suivante. Tout d'abord, si une métaséquence est vide, p est ajouté dans un nouvel itemset de la métaséquence. Si la métaséquence n'est pas vide (car au moins un autre motif d'action y a été ajouté auparavant), alors la métaséquence est parcourue et p est comparé avec chaque motif d'action p_y de la métaséquence successivement, pour savoir s'il doit être inséré avant, dans le même itemset ou après. Pour effectuer cette comparaison, l'implémentation utilise l'information que chaque motif d'action est extrait à partir d'une base de MD-séquences d'actions correspondant à une paire d'états. Si la paire d'états de p apparaît avant celle de p_y dans la séquence d'états correspondant à la métaséquence, p est ajouté dans un nouvel itemset inséré avant celui contenant p_y . S'ils apparaissent dans la même paire d'états, p est ajouté à l'itemset contenant p_y . Sinon, la boucle continue. À la fin de la boucle, si p n'a pas été inséré, il est ajouté à la fin de la métaséquence dans un nouvel itemset.

Après l'insertion de tous les motifs d'actions dans la métabase de séquences, des règles séquentielles peuvent être extraites entre les motifs d'actions en appliquant CMRules ou CMDeogun. La figure 5.13 résume comment la création de la métabase et l'extraction des règles séquentielles s'inscrivent dans l'approche d'acquisition automatique du chapitre 4.

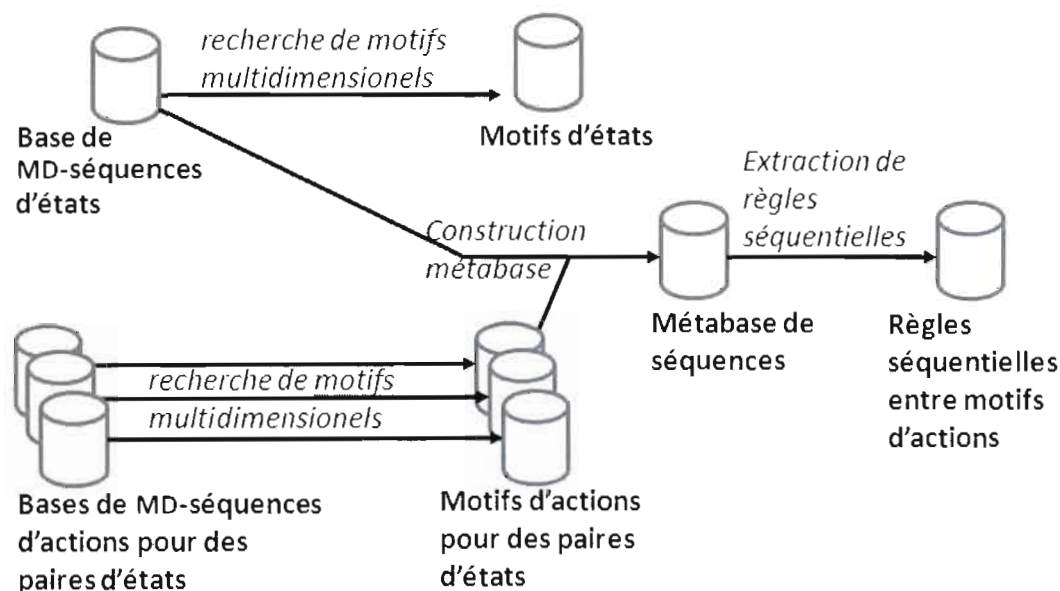


Figure 5.13 Processus d'acquisition des connaissances avec l'ajout des règles

5.7.2 Évaluation de la performance

Des expériences ont été effectuées pour évaluer la performance de la construction de la métabase et de la recherche de règles séquentielles avec CMRules et CMDeogun dans CanadarmTutor.

Pour ces expériences, les 7388 motifs d'actions découverts dans les 170 traces d'utilisations enregistrées pour un exercice de déplacement de Canadarm2, présentées à la section 4.9, ont été utilisés. Construire la métabase de séquences pour ces motifs a pris 4.12 secondes. Elle est constituée de 170 séquences contenant en moyenne 274.74 itemsets (écart-type de 148.43 et maximum de 735) et 349.69 items (écart-type de 176.91 et un maximum de 911). Les algorithmes CMRules et CMDeogun ont été appliqués sur cette métabase avec différents paramètres. Ces expériences sont décrites dans les prochains paragraphes, puis des conclusions sont tirées.

Lors de la première expérience, CMDeogun et CMRules ont été exécutés avec $minLeft = 1$, $maxLeft = \infty$, $minLeft = 1$, $maxRight = \infty$, $minSeqSup = 0.5$ et $minSeqConf =$

0.8. Un *minSeqConf* élevé a été choisi pour avoir des règles avec une forte confiance, ce qui est approprié dans le contexte de CanadarmTutor. 89203 règles ont été trouvées, contenant en moyenne 7.45 items avec un maximum de 14. CMDeogun a été beaucoup plus performant que CMRules avec un temps d'exécution de 76 s plutôt que de 3746,53 s.

Lors d'une deuxième expérience, les paramètres *maxLeft* = 2, et *maxRight* = 2 ont été utilisés, pour extraire des règles plus petites. Lors de cet essai, CMDeogun a demeuré plus rapide que CMRules en prenant 1,25 s pour découvrir les 2167 règles valides, alors que CMRules en a pris 5,06 s. Le nombre d'items par règle était 3.52 (écart-type de 0.59 et maximum de 4). Cette expérience a été répétée pour *minSeqSup* = 0.4. Les temps d'exécution de CMDeogun et CMRules ont été respectivement 67,12 s et 71,82 s, et 52653 règles ont été trouvées (contenant en moyenne 3.76 items avec un écart-type de 0.44). Encore une fois CMDeogun a été plus rapide. Pour cette raison seul CMDeogun a été exécuté pour un autre essai avec *minSeqSup* = 0.35. CMDeogun a pris 36,74 s pour découvrir les 199939 règles valides (contenant en moyenne 3.84 items avec un écart-type de 0.37).

Finalement, CMRules et CMDeogun ont été exécutés avec *minLeft* = 1, *maxLeft* = 1, *minLeft* = 1, *maxRight* = 1, *minSeqSup* = 0.1 et *minSeqConf* = 0.8 pour extraire des règles de taille 1 * 1, seulement. CMDeogun et CMRules ont pris respectivement 98,57 s et 180,71 s pour trouver les 27552 règles valides.

De ces expériences les conclusions suivantes sont tirées. Premièrement, le coût de la construction de la métabase est négligeable (4.12 secondes) par rapport à celui de l'extraction des règles. Deuxièmement, l'extraction des règles est plus coûteuse que la recherche de motifs séquentiels (cf. section 4.9). Troisièmement, cette expérience a montré l'importance de bien choisir *minSeqSup*, *maxLeft* et *maxRight*, car ces trois facteurs ont une forte influence sur le nombre de règles trouvées et sur la performance. Quatrièmement, dans toutes ces expériences CMDeogun a été plus performant que CMRules.

5.7.3 Amélioration des services tutoriels

Les règles séquentielles extraites peuvent être exploitées par les services tutoriels décrits au chapitre précédent. Dans CanadarmTutor, les règles ont permis d'améliorer le service tutoriel de suggestion d'indices de CanadarmTutor (cf. section 4.8). Ce service tutoriel consiste à présenter des motifs d'états à l'apprenant dans la fenêtre décrite à la figure 4.13 lorsqu'il demande un indice en cliquant sur « What next ? ». Quand l'apprenant sélectionne un motif d'état dans cette fenêtre, la liste des motifs d'actions pour passer du premier état au suivant est affichée. L'apprenant peut ensuite utiliser des boutons pour demander l'affichage des motifs d'actions pour les états suivants.

Pour améliorer ce service, CanadarmTutor a été modifié pour noter au cours d'un exercice les motifs d'actions suivis complètement par l'apprenant (détectés par RecognizePlan). Ensuite, la fenêtre a été altérée pour que les motifs d'actions apparaissant dans le conséquent d'une règle telle que tous les motifs d'actions de son antécédent ont été suivis par l'apprenant soient affichés en caractère gras. Cette nouvelle fenêtre est illustrée à la figure 5.14. Elle permet de mettre en évidence les motifs d'actions qui sont les plus probables d'être suivis par l'apprenant sachant les motifs d'actions qu'il a accompli.

Le service tutoriel d'exploration libre des motifs (cf. section 4.8.3) et d'évaluation du profil de l'apprenant (cf. section 4.8.1) n'ont pas été modifiés pour exploiter les règles, car cela semblait moins pertinent que pour la suggestion d'indices, mais des améliorations sont possibles. Par exemple, pour l'exploration libre des motifs, la navigation des règles pourrait être ajoutée. Pour l'évaluation du profil de l'apprenant, une stratégie plus complexe de mise à jour du modèle de l'apprenant pourrait être employée. Par exemple, les indicateurs de maîtrise du modèle de l'apprenant pourraient être augmentés ou diminués davantage quand un motif apparaît dans le conséquent d'une règle dont l'antécédent est vérifié, même s'il n'est suivi que partiellement.

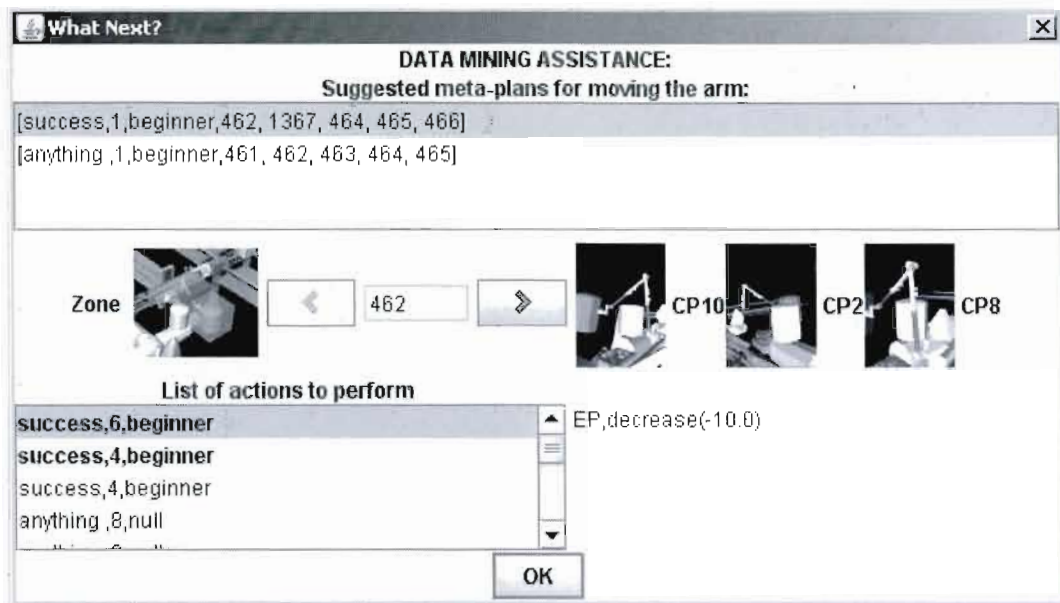


Figure 5.14 Nouvelle fenêtre pour la suggestion d'indice

5.8 Généricité de la solution

Les contributions présentées dans ce chapitre peuvent être réutilisées dans le domaine des STI et en fouille de données.

5.8.1 Généricité de l'approche d'acquisition automatique pour d'autres STI

L'approche d'acquisition automatique enrichie avec les règles séquentielles peut être appliquée pour d'autres tâches satisfaisant les contraintes présentées à la section 4.11.1 du chapitre précédent. L'ajout des règles séquentielles n'a aucun impact sur l'applicabilité de l'approche pour une tâche, car la découverte de règles est un posttraitement sur les motifs découverts.

5.9 Généricité des algorithmes CMRules et CMDeogun en fouille de données

L'utilité des algorithmes CMRules et CMDeogun pour d'autres applications a été démontrée par l'application de CMRules dans un second projet. Ce projet est un agent cognitif nommé CTS [41, 47] (aussi présenté brièvement à la section 4.11.2 de cette thèse) construit pour offrir une assistance à des apprenants lors d'activités d'apprentissage dans des environnements virtuels. CTS a été intégré dans CanadarmTutor indépendamment du travail de cette thèse pour tester des dialogues tutoriels. Pendant des sessions d'apprentissage, CTS peut interagir avec des apprenants par des dialogues comprenant indices, questions et explications. CTS prend des décisions pédagogiques sur la base de règles encodées dans une structure nommée le *réseau des actes*, créé à la main [41, 47].

L'algorithme CMRules a été intégré dans CTS pour lui donner la capacité d'apprendre des relations entre les événements et ainsi doter CTS d'un mécanisme fonctionnel d'apprentissage causal [48]. Les prochains paragraphes décrivent succinctement cette intégration sous l'angle de la fouille de données, car l'aspect applicatif sera détaillé dans la thèse de Usef Faghihi.

Pour intégrer l'algorithme CMRules à CTS, CTS a tout d'abord été modifié pour qu'il enregistre chacune de ses exécutions sous la forme d'une séquence contenant les informations perçues par CTS et les décisions qu'il a prises, ordonnées en fonction du temps. Ensuite, CTS a été modifié pour qu'il exécute après chaque exécution l'algorithme CMRules pour trouver les règles séquentielles communes à ses exécutions.

Une expérience a été réalisée dans laquelle 100 exécutions de CTS ont été effectuées. Pendant ces exécutions, les questions demandées par CTS ont été répondues de façon plausible. Ceci a résulté en une base de 100 séquences contenant en moyenne 25 itemsets. L'algorithme a été exécuté à partir de la quatrième exécution avec $minSeqSup = 0.40$ et $minSeqConf = 0.60$. Les règles séquentielles représentaient en moyenne 31.05 % des règles d'association et le nombre moyen de règles séquentielles était 27. Le temps d'exé-

cution de l'algorithme est demeuré court (moins de 50 ms). Plusieurs règles pertinentes pour CTS ont été découvertes telles que « l'apprenant effectue une proximité avec la station spatiale \Rightarrow l'apprenant déplace le mauvais joint » et « l'apprenant est inactif \Rightarrow l'apprenant manque de motivation ».

Pour exploiter les règles découvertes, CTS a été doté d'un mécanisme pour faire des prédictions et altérer son comportement en fonction de celles-ci. Le mécanisme procède en vérifiant si les parties gauches des règles sont applicables à l'exécution courante de CTS. Si plusieurs règles sont applicables, CTS choisit la règle r avec le meilleur produit $seqSup(r) \times seqConf(r)$. Les faits dans la partie droite de la règle choisie sont considérés déduits. Par exemple, si la règle « l'apprenant est inactif \Rightarrow l'apprenant manque de motivation » s'apparie avec l'exécution courante qui contient « l'apprenant est inactif », CTS déduira que cela est causé par un manque de motivation.

De façon globale, il a été observé que le comportement de CTS s'est adapté au fil des exécutions en fonction des règles découvertes de façon positive [48]. Par contre, il n'a pas été mesuré empiriquement comment les dialogues tutoriels offerts par la nouvelle version de CTS favorisent l'apprentissage. Le lecteur pourra se référer à un article décrivant l'application de l'algorithme CMRules dans CTS pour plus de détails [48] et à la thèse de Usef Faghihi.

5.10 Conclusion

Le chapitre précédent a présenté une approche pour l'acquisition automatique des connaissances du domaine dans les STI pour les domaines procéduraux et mal définis. Cette approche repose sur l'idée de diviser un problème en sous-problèmes appelés *états* et d'extraire (1) des séquences fréquentes d'états visités et (2) des séquences fréquentes d'actions pour passer de chaque état à chaque état adjacent (voir section 4.6). Toutefois, un problème subsistait avec cette solution : les motifs d'actions trouvés pour une paire d'états n'étaient pas liés avec ceux trouvés pour une autre paire d'états. Il a donc été décidé d'extraire des motifs temporels entre les motifs pour les lier entre eux et ainsi

obtenir un espace de motifs plus structuré pour supporter des services tutoriels enrichis.

Le chapitre a présenté une revue de littérature de plusieurs approches en fouille de données pour extraire des motifs temporels pour faire des prédictions, et plus particulièrement les principaux travaux sur les règles séquentielles. Toutefois, ces travaux ne sont pas adaptés pour la découverte de règles communes à plusieurs séquences. Le chapitre a donc défini le problème de la découverte de règles séquentielles communes à plusieurs séquences.

Le chapitre a ensuite présenté deux algorithmes pour ce problème : CMRules et CMDeogun. Parce que les algorithmes proposés ont une fin différente de celles des algorithmes de la littérature, ils n'utilisent pas une fenêtre temporelle glissante comme plusieurs autres le font. L'algorithme CMRules cherche plutôt des règles d'association entre items pour réduire l'espace de recherche à des items survenant ensemble dans plusieurs séquences. Ensuite, l'algorithme élimine les règles d'association qui ne respectent pas *minSeqSup* et *minSeqConf*. Ce chapitre a prouvé que ce processus génère correctement l'ensemble des règles séquentielles si les paramètres pour l'extraction de règles d'association sont $minsup = minSeqSup$ et $minconf = minSeqConf$.

Que CMRules soit basé sur l'extraction des règles d'association a deux autres bénéfices. Tout d'abord, cette approche est très efficace si quelqu'un désire découvrir à la fois des règles d'association et des règles séquentielles dans une base de séquences, puisque les deux sont découvertes en même temps. Deuxièmement, il est possible de donner des capacités supplémentaires à l'algorithme CMRules en choisissant un algorithme d'extraction de règles d'association possédant cette capacité (cf. section 5.4.5).

Le deuxième algorithme, CMDeogun, est une adaptation d'un algorithme de la littérature. Il a été adapté pour extraire des règles communes à plusieurs séquences et aussi pour trouver l'ensemble complet des règles (l'algorithme original ne trouvait qu'un sous-ensemble). L'algorithme CMDeogun utilise une approche différente de CMRules en trouvant d'abord des règles avec un antécédent et un conséquent de taille 1 puis en les combinant récursivement pour générer des règles plus grandes. Quatre propriétés ont

été présentées pour élaguer l'espace de recherche. Le chapitre a aussi fait un survol des possibilités d'extension de CMDeogun.

La performance des algorithmes CMRules et CMDeogun a été évaluée de plusieurs façons. Premièrement, la complexité en temps d'exécution de CMRules a été analysée. Deuxièmement, la performance des deux algorithmes a été comparée avec trois bases de séquences publiques d'applications réelles. Avec cette expérience, il a été observé empiriquement que les deux algorithmes sont chacun plus performants dans certains cas. La faiblesse de CMRules est que son temps d'exécution croît proportionnellement au nombre de règles d'association. Les faiblesses de CMDeogun sont qu'il génère parfois une grande proportion de règles candidates invalides (des règles ne satisfaisant pas les critères *minSeqSup* et *minSeqConf*), ce qui augmente le coût de la découverte des règles valides, et qu'il réussit moins à profiter du seuil *minSeqConf* pour améliorer sa performance.

Le chapitre a ensuite montré comment CMRules et CMDeogun ont été intégrés dans l'approche d'acquisition automatique pour les STI présentée au chapitre précédent. Une procédure pour construire une métabase de séquences pour l'extraction de règles séquentielles entre motifs d'actions a été proposée. Puis des tests de performance avec CanadarmTutor et l'adaptation du service tutoriel de génération d'indices ont été décrits.

Ensuite, la généralité de l'approche d'acquisition automatique pour d'autres tâches a été discutée. Puis, pour illustrer l'utilité de CMRules et CMDeogun à d'autres problèmes de fouille de données, une application dans un agent cognitif nommé CTS a été présentée. Grâce à l'intégration de CMRules, l'agent peut dorénavant apprendre des relations temporelles entre les événements qui sont communes à chacune de ses exécutions. La performance de l'algorithme CMRules dans cette application s'est montrée très satisfaisante.

Pour des travaux futurs, des algorithmes alternatifs pourraient être conçus. En particulier, l'auteur de cette thèse prévoit explorer la possibilité d'extraire des représentations

compactes et sans perte d'information de l'ensemble des règles séquentielles et l'utilisation d'autres mesures d'intérêt pour les règles.

CHAPITRE VI

UN MODÈLE HYBRIDE POUR LES DOMAINES MAL DÉFINIS

Trois approches ont été expérimentées pour donner une expertise du domaine à CanadarmTutor : un planificateur de trajectoire (mentionné à la section 1.5.3) [19, 93], un modèle cognitif (présenté au chapitre 2) et une approche d'acquisition automatique de connaissances du domaine (présentée aux chapitres 4 et 5). Chacune de ces approches a certains avantages et certaines limites.

Le planificateur est capable de générer une trajectoire du bras évitant les obstacles entre n'importe quelle configuration du bras canadien. Cela permet de supporter des services tutoriels comme la présentation de démonstrations aux apprenants. Par contre, les trajectoires générées ne sont pas toujours réalistes ou faciles à suivre, puisqu'elles ne sont pas basées sur l'expérience humaine, et qu'elles ne couvrent pas d'autres aspects de la tâche de manipulation de Canadarm2 tel que sélectionner les caméras et ajuster leurs paramètres. De plus, le planificateur ne peut pas supporter des services tutoriels importants comme l'évaluation des connaissances d'un apprenant, puisqu'il n'a pas de représentations des connaissances.

Le modèle cognitif pallie certaines limites du planificateur. Il permet d'évaluer de façon détaillée les connaissances nécessaires pour effectuer les principales étapes du déplacement du bras robotisé. Il a été conçu pour évaluer le raisonnement spatial, crucial à cette tâche. Néanmoins, il ne permet pas de descendre au niveau de détails fin de comment choisir les rotations de joints à effectuer. La raison est qu'il n'y a pas de stratégie claire,

complète et facilement formalisable pour déterminer les rotations de joints à effectuer pour atteindre une configuration donnée (il n'y en a pas d'enseigner à l'Agence Spatiale Canadienne [40]). Un opérateur doit en réalité tenir compte de plusieurs critères difficiles à définir comme la sécurité des manoeuvres et sa familiarité avec celles-ci (ce ne sont pas les plus courtes en distance ou en temps qui sont les meilleures, mais les plus sécuritaires), des habiletés acquises par la pratique (cf. section 1.5.2). Aussi, une limite de l'utilisation du modèle cognitif est que créer un modèle de tâches à la main est coûteux en temps.

L'approche automatique d'acquisition de connaissances du domaine présentée aux chapitres 4 et 5 répond en grande partie aux limites du planificateur et du modèle cognitif, en permettant d'offrir plusieurs services tutoriels comme la recommandation de rotations de joints, et contrairement au planificateur, cette assistance est basée sur l'expérience humaine. Cependant, une limite de cette approche est qu'aucune aide ne peut être offerte à un apprenant s'il explore une solution qui n'est pas couverte par les motifs extraits.

Sur la base de la constatation des limites et avantages de chaque approche, il a été décidé de construire un modèle hybride avec les trois approches pour tirer profit du meilleur de chacune dans CanadarmTutor. Ceci est en accord avec Lynch et al. [105] qui ont énoncé, dans un article faisant une revue de littérature sur la problématique des domaines mal définis, que les approches hybrides sont les plus prometteuses pour construire des STI pour ces domaines.

Ce chapitre est organisé de la façon suivante. Il présente en premier lieu le modèle hybride implémenté dans CanadarmTutor. Ensuite, les deux sections suivantes décrivent respectivement le cadre méthodologique et les résultats d'une évaluation empirique de la nouvelle version de CanadarmTutor. Finalement une conclusion du chapitre est présentée.

6.1 Le modèle hybride

Le modèle hybride a été construit avec pour objectif d'intégrer les services tutoriels offerts par chaque approche en un tout cohérent. Cette section décrit le modèle hybride par une description des différents services tutoriels qu'il permet d'offrir.

6.1.1 Évaluation du profil de l'apprenant

Le premier service tutoriel est l'évaluation du profil de l'apprenant. Cette évaluation est réalisée en partie avec le modèle cognitif et en partie avec les motifs découverts par l'approche d'acquisition automatique.

Évaluation avec le modèle cognitif

L'évaluation avec le modèle cognitif est réalisée comme décrite au chapitre 2.

Pour les exercices procéduraux de déplacement du bras canadien, un module nommé le « traceur de modèle » (cf. section 2.5) compare les actions faites par l'apprenant avec celles prescrites par le modèle cognitif pour évaluer les connaissances de l'apprenant. Le traceur de modèle détecte deux types d'erreurs : les erreurs procédurales et les erreurs de rappel de connaissances sémantiques. Au fil d'un exercice, le traceur de modèle met à jour les indicateurs de maîtrise du modèle de l'apprenant (cf. section 2.7.2). Un indicateur de maîtrise est présent pour chaque connaissance du modèle cognitif et prend une valeur entre zéro et un.

Le modèle cognitif est aussi utilisé pour évaluer les connaissances sémantiques dans le cadre de questions posées à l'utilisateur (cf. section 2.7.3). Cinq types de questions sont implémentés dans CanadarmTutor. Chaque fois qu'un apprenant répond correctement ou incorrectement à une question, les indicateurs de maîtrise des connaissances liées à cette question sont mis à jour (cf. section 2.7.3).

Évaluation avec les motifs découverts par l'approche d'acquisition automatique

L'évaluation du profil de l'apprenant par l'analyse des motifs qu'il suit a été présentée à la section 4.8.1. Le prochain paragraphe résume son fonctionnement, puis le paragraphe suivant mentionne les changements apportés dans le cadre du modèle hybride.

Lors d'un exercice procédural, après chaque action effectuée par l'apprenant, l'algorithme `RecognizePlan` est appliqué pour déterminer les motifs qui s'apparient avec les dernières actions de l'apprenant (cf section 4.8). Pour chaque motif suivi, le service tutoriel d'évaluation met à jour les indicateurs de maîtrise correspondants à chaque valeur de dimension du motif (cf. 4.8.1). Initialement, seulement deux dimensions avaient été prévues pour annoter les traces d'utilisation dans `CanadarmTutor`. La première, « État de la solution », indique si une trace s'est terminée par la complétion de l'exercice avec succès. La seconde, « niveau d'expertise », indique le niveau d'expertise de la personne ayant enregistré la trace (débutant, novice ou expert).

Pour le modèle hybride, il a été décidé d'ajouter de nouvelles dimensions aux traces d'utilisation et par conséquent aux motifs découverts. Chaque trace est dorénavant annotée avec des dimensions qui correspondent à des connaissances du modèle cognitif. Puisque toutes les connaissances du modèle cognitif ne sont pas pertinentes pour tous les exercices et qu'ajouter des dimensions peut alourdir le processus de fouille de données, un sous-ensemble de connaissances a été choisi manuellement pour annoter les traces pour chaque exercice. Par exemple, pour l'exercice décrit à la section 4.9, 23 connaissances ont été choisies pour annoter les traces en plus des dimensions « Niveau d'expertise » et « État de la solution ». Ces dimensions additionnelles sont la connaissance de 6 caméras jugées pertinentes pour l'exercice, la connaissance des 16 principaux sous-espaces (cf. section 2.6) et la procédure complexe pour déplacer le bras canadien. Pour une trace, chacune de ces dimensions prend soit la valeur « maîtrisée » ou « non-maîtrisée » pour indiquer si la personne qui a effectué la trace maîtrisait la connaissance. Le service tutoriel d'évaluation peut ensuite évaluer si l'apprenant possède chaque connaissance en

regardant les motifs suivis, de la même façon que pour les dimensions « État de la solution » et « Niveau d'expertise ». Dans le modèle de l'apprenant, une distinction est faite entre l'évaluation réalisée avec les motifs et celle faite par le traceur de modèle. Ainsi, chaque connaissance du modèle cognitif possède deux indicateurs de maîtrise (une pour l'évaluation avec les motifs et une pour l'évaluation réalisée par le traceur de modèle). Pour l'enregistrement des traces, seuls les indicateurs calculés par le modèle cognitif sont utilisés afin d'éviter que l'évaluation réalisée avec les motifs annote les traces et influence éventuellement les prochains motifs générés (ce qui pourrait faire une boucle de renforcement). Pour les services tutoriels présentés dans les prochaines sous-sections, l'indicateur de maîtrise utilisé pour chaque connaissance est la moyenne des deux indicateurs.

6.1.2 Génération d'exercices personnalisés

La génération d'exercices personnalisés est réalisée tel qu'expliquée à la section 2.7.5. Après l'accomplissement de plusieurs exercices par l'apprenant, CanadarmTutor acquiert un portrait détaillé des connaissances que l'apprenant maîtrise par le processus d'évaluation des connaissances décrit à la section 6.1.1. Le système utilise cette information pour générer des questions et choisir des exercices procéduraux qui progressivement impliquent de nouvelles connaissances ou des connaissances jugées non maîtrisées.

6.1.3 Génération d'indices

La génération d'indices consiste à suggérer à l'apprenant les prochaines actions à effectuer dans un exercice procédural. Ce service tutoriel est déclenché quand l'apprenant clique sur le bouton « What Next ? » dans l'interface de CanadarmTutor. Deux versions de ce service tutoriel ont été présentées aux sections 2.7.4 et 4.8.2 utilisant respectivement le modèle cognitif et les motifs. Une troisième version existe avec le planificateur de trajectoire [19, 93, 124]. Ces trois versions sont complémentaires, car la première renseigne l'apprenant sur la procédure générale de déplacement du bras robotisé, la seconde l'assiste au niveau des rotations de joints et la troisième illustre des trajectoires possibles. Par conséquent, pour le modèle hybride, elles ont été intégrées ensemble. La

figure 6.1 illustre la nouvelle version de ce service tutoriel.

Dans la partie (A) de la fenêtre principale, le modèle cognitif indique sous forme de texte la prochaine étape de la procédure générale pour déplacer le bras robotisé. Par exemple, à la figure 6.1, le message « You should select a camera and then adjust its parameters for monitor 2 » est affiché à l'apprenant. Ces consignes générales sont générées comme expliqué à la section 2.7.4, grâce au traçage de modèle avec le modèle cognitif.

La partie (B) de la fenêtre principale liste les motifs s'appariant avec la solution de l'utilisateur. Ils indiquent les rotations de joints pouvant être faits dans sa situation pour déplacer le bras robotisé. Deux types de motifs sont accessibles : les motifs d'états et les motifs d'actions. Cette section de la fenêtre est identique à celle présentée à la section 5.7.3.

Finalement, si aucun motif ne s'apparie avec la solution de l'apprenant, une démonstration est générée par le planificateur de trajectoire pour illustrer une solution possible. Cette trajectoire est présentée à l'apprenant dans une seconde fenêtre illustrée dans la partie (C) de la figure 6.1 (voir les articles sur le planificateur de trajectoire pour des détails sur son fonctionnement [19, 93, 124]).

6.1.4 Génération de conseils

Un autre service tutoriel est la présentation de conseils personnalisés à l'apprenant (cf. section 2.7.4). Lorsque l'apprenant clique sur « Advices », un ensemble de conseils personnalisés lui sont présentés. Ces conseils portent sur les connaissances les moins bien maîtrisées selon le modèle de l'apprenant. Les conseils sont des messages textuels extraits des annotations des connaissances du modèle cognitif.

6.1.5 Exploration libre des motifs et connaissances et démonstrations

L'apprenant peut explorer librement les motifs séquentiels pour un exercice procédural (cf. section 4.8.3). Il peut aussi explorer le modèle cognitif pour apprendre la procédure

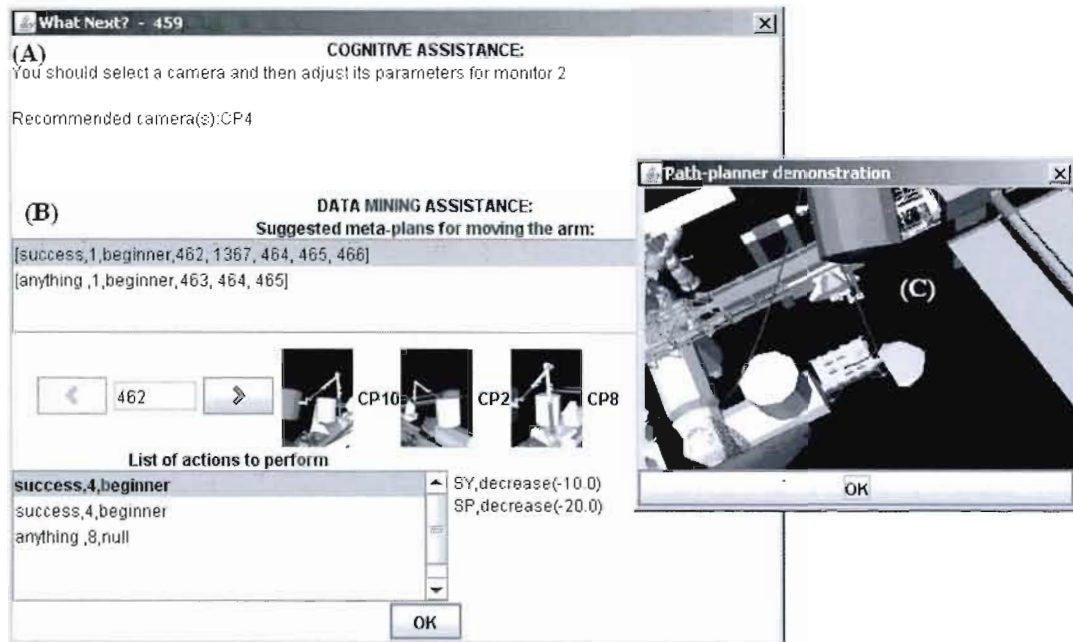


Figure 6.1 Génération d'indices par le modèle hybride

générale de déplacement du bras robotisé et se familiariser avec les connaissances sémantiques importantes (cf. section 2.7.1). Aussi, il peut demander des démonstrations du modèle cognitif (cf. section 2.7.4) ou du planificateur de trajectoire.

6.1.6 Rétroaction immédiate et aide proactive

Les deux derniers services tutoriels sont la rétroaction immédiate et l'aide proactive. Ces services sont offerts sur la base du traçage de modèle (cf. section 2.7.6). Ils sont activés ou désactivés par l'apprenant. La rétroaction immédiate consiste à fournir une rétroaction à l'utilisateur sous forme d'un message après chaque erreur procédurale pendant les exercices de déplacement de Canadarm2. L'aide proactive consiste à indiquer à l'apprenant par un message lorsqu'il devrait changer une ou plusieurs caméras pour obtenir une meilleure vue de la scène d'opération.

6.2 L'évaluation empirique : cadre méthodologique

Pour évaluer les contributions de cette thèse, il a été décidé de faire une analyse qualitative fine avec quelques utilisateurs de CanadarmTutor plutôt que de faire une évaluation empirique avec de nombreux utilisateurs. Ce type d'évaluation a été choisi, car il était plus approprié pour être réalisé dans les échéances du doctorat, sachant qu'une expérimentation plus large est prévue comme suite au doctorat.

Dans le contexte de l'évaluation d'un logiciel, l'analyse qualitative fine est une analyse détaillée de l'utilisation du logiciel par un petit nombre de participants. Les bénéfices d'employer cette approche sont (1) d'analyser en détails chaque cas, ce qui n'est souvent pas envisageable avec un très grand nombre de participants, et (2) d'être relativement simple à mettre en place. Toutefois, le désavantage de cette approche est que bien que des tendances peuvent être observées, il est difficile d'en tirer des généralités, la taille des échantillons n'étant pas significative statistiquement.

6.2.1 Les hypothèses

Les hypothèses retenues pour l'expérimentation portent sur deux aspects. Le premier aspect est de savoir si les services tutoriels favorisent l'apprentissage, c'est-à-dire s'ils aident les apprenants à corriger leurs erreurs ou à améliorer leur performance au fil d'une séance d'utilisation. Le deuxième aspect est de savoir si les rétroactions offertes par les services tutoriels sont généralement pertinentes dans leur contexte. Il y a six hypothèses correspondant à ces deux aspects pour les trois grandes contributions de la thèse :

HC1 : l'aide fournie par le modèle cognitif favorise l'apprentissage.

HC2 : l'aide fournie par le modèle cognitif est pertinente.

HA1 : l'aide fournie par l'approche d'acquisition automatique favorise l'apprentissage.

HA2 : l'aide fournie par l'approche d'acquisition automatique est pertinente.

HH1 : l'aide fournie par le modèle hybride favorise l'apprentissage.

HH2 : l'aide fournie par le modèle hybride est pertinente.

Les autres aspects comme l'efficacité des algorithmes de fouille de données ont été discutés dans les chapitres précédents, et n'ont donc pas été pris en compte dans cette expérience.

6.2.2 La conditions étudiée

Pour tester les hypothèses, il a été décidé d'effectuer seulement une expérimentation avec la version de CanadarmTutor intégrant le modèle hybride. Cette décision a été prise pour simplifier l'expérimentation et parce que la plupart des services tutoriels de l'approche d'acquisition automatique et du modèle cognitif se retrouvent intégrés dans le modèle hybride sans grande modification. Le planificateur de trajectoire a été désactivé, car il n'est pas une contribution de la thèse.

Pour fournir des motifs aux services tutoriels de l'approche d'acquisition automatique, 30 traces d'utilisation ont été enregistrées pour chaque exercice préalablement à l'expérience par l'auteur de cette thèse. La seule exception est l'exercice décrit à la section 4.9 pour lequel 170 traces ont été enregistrées. Pendant l'expérience, l'apprentissage de nouveaux motifs a été désactivé afin d'assurer la même base de motifs pour tous les apprenants et ainsi simplifier l'analyse des résultats.

6.2.3 Les personnes impliquées

L'expérience implique deux *participants* et un *expérimentateur*, qui est l'auteur de cette thèse. Les participants ont été recrutés au département d'informatique de l'Université du Québec à Montréal par l'entremise de professeurs. Ils sont des étudiants en informatique de deuxième ou troisième cycle qui n'ont jamais travaillé avec un bras robotisé. Le premier critère est pour s'assurer que les participants aient des niveaux de compétence en informatique similaires. Le deuxième critère est pour s'assurer que tous les participants soient débutants par rapport à la tâche. Les participants ont été recrutés sur une base

volontaire et n'ont pas été récompensés.

6.2.4 La procédure

La procédure suivante a été mise en oeuvre pour chaque participant. Premièrement, l'expérimentateur s'assure que le participant a signé le formulaire de consentement et qu'il en a pris connaissance. Puis l'expérimentateur rappelle au participant les objectifs de l'expérimentation, la procédure et les données qui seront recueillies.

Deuxièmement, l'expérimentateur présente CanadarmTutor et les principes de base de la manipulation de Canadarm2 au participant.

Troisièmement, le participant utilise le logiciel pour faire 15 exercices procéduraux. Pendant ces exercices, il peut accéder à tous les services tutoriels sauf le « Space Quiz » (cf. section 2.7.3), car ce dernier est plus approprié pour un long entraînement. Pendant l'utilisation, le logiciel enregistre pour chaque exercice (1) si l'exercice est réussi, (2) la séquence d'actions faites par l'apprenant pour accomplir l'exercice, (3) les moments où l'apprenant a fait appel aux services tutoriels. L'expérimentateur observe l'utilisation pour (1) évaluer la pertinence des recommandations du logiciel dans leur contexte, et (2) observer si les conseils du logiciel ont amené le participant à améliorer sa performance ou s'ils lui ont nui. Pour réaliser le premier objectif, l'expérimentateur note si l'assistance semble pertinente dans le contexte chaque fois que l'apprenant utilise un service tutoriel. Pour accomplir le second objectif, l'expérimentateur note les occurrences où un service tutoriel semble avoir aidé ou nui à l'apprenant pour améliorer sa performance dans des exercices subséquents, par exemple en corrigeant une erreur qu'il commettait régulièrement. Toutes les observations sont faites sur papier avec une grille d'observation.

Quatrièmement, l'expérimentateur effectue une entrevue d'une durée de quinze minutes avec l'apprenant pour corroborer les observations. L'expérimentateur pose aussi deux questions à l'apprenant : « Est-ce que vous trouvez de façon générale que les services tutoriels sont pertinents? » et « Est-ce que vous trouvez de façon générale que les services tutoriels favorisent ou nuisent à votre apprentissage? » pour obtenir l'opinion

de l'apprenant. Toutes les réponses sont notées sur papier avec une grille d'entrevue.

Cinquièmement, l'expérimentateur analyse conjointement les résultats obtenus par l'observation, l'entrevue et les informations recueillies par le logiciel. Ensuite l'expérimentateur répète la procédure pour les autres participants.

6.2.5 Prise en compte des considérations éthiques

L'aspect éthique a été pris en compte dans la conception de l'expérience de plusieurs façons :

Anonymisation : Aucune information personnelle n'a été demandée aux participants, l'objectif de l'expérience étant d'évaluer le logiciel. Toutes les informations recueillies pendant l'expérience ont été anonymisées en identifiant les participants par un numéro. Le seul endroit où le nom de chaque participant apparaît est sur le formulaire de consentement, mais les noms ne sont pas associés à leur numéro.

Stockage des données : Toutes les informations recueillies ont été stockées dans un endroit sûr, sous clé. Deux ans après le dépôt de la version finale de la thèse et des dernières publications, les formulaires de consentement seront détruits, mais les données seront conservées.

Description détaillée de l'expérience : Lors du recrutement, l'expérimentateur a fourni une description détaillée de la recherche, son déroulement, ses objectifs et les données recueillies aux participants de façon à ce qu'ils puissent prendre une décision éclairée en choisissant ou non de participer.

Abandon sans pénalité : Les participants ont été informés qu'ils pouvaient abandonner l'expérience en tout temps sans pénalité et abandonner un exercice pour passer au suivant.

L'expérience a été présentée au comité d'éthique de la faculté des sciences de l'UQAM et a été approuvée le 27 avril 2010 avec quelques modifications mineures.

6.3 L'évaluation empirique : les résultats

L'expérience a été effectuée pendant les deux dernières semaines de mai 2010. Chaque participant a utilisé CanadarmTutor pendant environ 50 minutes, incluant quelques courtes pauses. Les entrevues ont duré environ cinq minutes chacune. Le temps pour accomplir chaque exercice a varié d'une à six minutes en fonction de la difficulté de l'exercice, de la solution choisie, de l'hésitation de l'utilisateur et de sa motivation à effectuer les déplacements avec précision et à respecter la procédure de déplacement du bras canadien. Aucun des participants n'a abandonné un exercice ou échoué à un exercice en effectuant une collision avec la station spatiale. Le premier participant a utilisé tous les services tutoriels. Le second les a tous utilisés sauf la rétroaction immédiate et l'aide proactive. L'utilisation des services tutoriels a varié grandement d'un exercice à l'autre : lors du premier exercice, les deux participants ont préféré travailler sans les services tutoriels pour essayer la tâche par eux-mêmes, puis ils ont progressivement exploré l'assistance disponible par les différents services tutoriels au cours des exercices subséquents.

6.3.1 La pertinence des services tutoriels

Lors de l'entrevue, les deux participants ont indiqué qu'ils trouvaient l'assistance fournie globalement pertinente. Cela correspond à ce qui a été observé de façon générale par l'expérimentateur. Les prochains paragraphes décrivent les remarques spécifiques pour chaque service tutoriel.

Génération d'exercices

Les deux utilisateurs ont trouvé en général que les exercices générés étaient pertinents. Le seul reproche des utilisateurs est que le même exercice peut être offert deux fois, ce qui est arrivé une fois pour chaque participant. Les utilisateurs ont indiqué qu'ils auraient préféré que le service tutoriel ne suggère pas un exercice deux fois avant d'avoir épuisé tous les autres exercices. L'algorithme de génération d'exercice (cf. section 2.7.5)

pourrait être adapté pour répondre à cette demande.

Génération d'indices

Aucun indice n'a été jugé non pertinent par les participants. Toutefois, le deuxième utilisateur a reproché qu'à deux reprises sur huit, aucun motif n'était disponible. Cette limite est une conséquence du faible nombre de traces enregistrées.

Génération de conseils

Tous les participants ont trouvé les conseils justes. Le premier participant a admis qu'il n'a demandé des conseils que deux fois parce qu'il n'était pas soucieux de suivre le protocole d'opération de Canadarm2. Le deuxième participant a consulté les conseils après chaque exercice pour voir s'il s'améliorait. Les conseils lui ont permis de corriger certaines erreurs commises pour l'ordre d'ajustement des caméras. L'appréciation des conseils est un indicateur de la qualité du modèle de l'apprenant.

Exploration libre des motifs

Le premier utilisateur a utilisé deux fois ce service et le second, une seule fois. Les deux utilisateurs ont convenu de sa pertinence, mais ont mentionné que l'interface utilisateur devrait être améliorée, ce qui a freiné l'usage.

Démonstrations

Les participants ont trouvé les démonstrations générées par le modèle cognitif pertinentes. Chaque participant a utilisé ce service une fois.

La rétroaction immédiate et l'aide proactive

Les services de rétroaction immédiate et d'aide proactive ont été utilisés seulement par le premier participant. Ce dernier les a utilisés pour un exercice, les a trouvés pertinents,

mais les a désactivés ensuite vu qu'il n'était pas intéressé à corriger toutes ses erreurs et désirait ne pas être trop souvent interrompu.

6.3.2 L'impact des services tutoriels sur l'apprentissage

Au niveau de l'apprentissage, les deux apprenants ont convenus que les services les ont aidés de façon générale. Le deuxième participant a nuancé que pour les exercices les plus faciles, l'aide n'est pas toujours nécessaire, mais que pour les exercices difficiles, elle est bénéfique.

6.3.3 Autres commentaires des participants

Hormis les commentaires sur l'apprentissage et la pertinence, les participants ont mentionné que la façon dont les motifs sont présentés et la convivialité de l'interface utilisateur en général pourrait être améliorée. En particulier, les deux participants ont indiqué qu'ils auraient préféré une interface plus intuitive pour consulter les motifs.

6.3.4 Conclusion de l'expérience

En somme, les services tutoriels offerts autant par le modèle cognitif que sur la base des motifs ont été jugés globalement pertinents et favorisant l'apprentissage, ce qui a été confirmé par les observations. Bien que les conclusions ne sont pas généralisables vu le petit nombre de participants, les résultats sont un indicateur positif. L'expérience a aussi permis d'identifier des lacunes de l'interface utilisateur même si cela n'était pas l'objectif de cette expérience. Elle pourront être corrigées avant d'effectuer une nouvelle expérience avec un plus grand nombre d'utilisateurs. Hormis l'utilisation d'un plus large échantillon et l'amélioration de l'interface, il serait préférable pour une future expérience d'avoir recours à des astronautes ou à des spécialistes de l'Agence Spatiale Canadienne pour obtenir un meilleur avis sur les services tutoriels.

6.4 Généricité du modèle hybride pour d'autres domaines

Le modèle hybride décrit dans ce chapitre est applicable pour d'autres domaines. Les conditions de son applicabilité sont les conditions propres à chacune des trois approches qu'il combine. Pour le modèle cognitif et l'approche d'acquisition automatique, elles ont été décrites respectivement aux sections 2.9 et 4.11.1. Pour le planificateur de trajectoire, ce sont les domaines incluant des tâches de déplacement d'objets ou de robots dans l'espace tel que la manipulation de robot pour construire des voitures.

Alternativement, si le modèle hybride n'est pas applicable à un domaine, une seule ou deux des trois approches peuvent être utilisées. De plus, un autre type de système expert pourrait remplacer le planificateur, ou d'autres approches de représentation des connaissances comme la modélisation à base de contraintes pourraient être employées.

6.5 Conclusion

Ce chapitre a d'abord rappelé les limites et avantages des trois approches testées dans CanadarmTutor (le modèle hybride, le planificateur et l'approche d'acquisition automatique). Ensuite, un modèle hybride a été proposé pour tirer profit du meilleur des trois approches et pallier leurs limites respectives. Ce modèle hybride n'est pas une simple addition des services offerts par chacune des trois approches, mais une intégration en un tout cohérent.

Le chapitre a décrit une évaluation empirique du modèle hybride réalisée avec deux utilisateurs. Pour cette expérience, le planificateur a été désactivé vu qu'il n'est pas une contribution de la thèse. Les conclusions de l'étude sont que les services tutoriels ont été appréciés, ont été pertinents et ont favorisé l'apprentissage. L'expérience a aussi permis d'identifier des améliorations à faire à l'interface utilisateur de CanadarmTutor. Ces améliorations sont prévues pour des travaux futurs, préalablement à la réalisation d'une évaluation avec davantage d'utilisateurs.

Pour clore, le chapitre a résumé les conditions pour appliquer le modèle hybride dans

des systèmes tutoriels intelligents conçus pour d'autres domaines.

CONCLUSION

Ce chapitre résume les contributions de la recherche, en note les limites et identifie des perspectives de travaux futurs.

Contributions de la thèse

Les contributions à la dimension cognitive du doctorat sont :

Un modèle hybride pour les STI. Un modèle hybride a été proposé pour répondre au problème de la représentation des connaissances des domaines procéduraux et mal définis dans les STI.

Un modèle cognitif pour le raisonnement spatial dans les STI. Le modèle cognitif est le premier à évaluer et à supporter le raisonnement spatial dans les STI, et plus généralement à évaluer le rappel des connaissances sémantiques dans les tâches procédurales. Le modèle a été appliqué dans CanadarmTutor pour fournir de l'assistance pour l'apprentissage de la manipulation de Canadarm2. Toutefois, il n'y est pas limité, car le modèle est basé sur les architectures cognitives ACT-R et MIACE qui se veulent génériques (ACT-R a d'ailleurs été utilisé pour modéliser plus d'une centaine de tâches [38]).

Une approche d'acquisition automatique L'approche d'acquisition automatique proposée se distingue des autres (cf. chapitre 3) en étant la seule (1) à être conçue pour les tâches procédurales où il y a un grand nombre de possibilités et aucune stratégie claire pour les réaliser et (2) à tenir compte du profil de chaque démonstrateur. De plus, l'approche a les propriétés intéressantes suivantes, contrairement à d'autres approches (cf. chapitre 3) : elle (1) ne requiert aucune spécification préalable des connaissances du domaine, (2) peut enrichir sa base de connaissances après chaque

résolution de problème par un apprenant, (3) ne suppose aucune vision restrictive de l'intelligence comme des règles de production et (4) ne demande pas aux démonstrateurs de fournir des informations supplémentaires comme leurs intentions lors des démonstrations.

Des moyens pour étudier la cognition. L'approche d'acquisition automatique peut être utilisée hors du contexte des STI pour étudier la cognition, car les modèles de tâches partiels permettent de mettre en lumière le processus de résolution de problèmes d'humains.

Les contributions à la dimension informatique du doctorat sont :

Une implémentation du modèle hybride intégrée à CanadarmTutor. Des structures de connaissances et des algorithmes pour les exploiter ont été implémentés et intégrés au logiciel CanadarmTutor.

Des algorithmes originaux de fouille de données. La thèse a proposé un algorithme de recherche de motifs séquentiels combinant plusieurs propriétés d'autres algorithmes et aussi de nouvelles comme le regroupement automatique de valeurs. Deux algorithmes ont aussi été proposés pour l'extraction de règles séquentielles (CM-Rules et CMDeogun). Tous ces algorithmes peuvent être réutilisés pour d'autres tâches de fouille de données qui ne sont pas nécessairement liées au domaine des STI. Les algorithmes qui ont été développés ont d'ailleurs déjà été appliqué dans le projet de l'agent cognitif CTS développé à l'UQAM pour créer un mécanisme d'apprentissage épisodique [47, 49, 55, 59] et un mécanisme d'apprentissage causal [48] (ces travaux ont été mentionnés dans les sections 4.11.2 et 5.9, respectivement).

Publications

Les contributions de la thèse ont été présentées dans plusieurs publications (chapitres de livre, articles de revues, actes de conférences internationales et atelier). Ce tableau regroupe ces publications en fonction de la communauté scientifique et de la contribution :

Contribution	Communauté scientifique			
	STI et e-learning	educational data mining	fouille de données	intelligence artificielle
modèle hybride	[68, 72]			
acquisition automatique - algo. motifs séquentiels - algo. règles séquentielles	[64, 67, 55, 125]	[70]	[69]	[65, 59, 71, 126]
modèle cognitif	[63, 60]			[62]
applications des algo. à d'autres domaines				[47, 49, 48]

Logiciel

De plus, depuis 2008, l'auteur de la thèse offre le code source de son implémentation de l'algorithme de recherche de motifs séquentiels ainsi que de 27 autres algorithmes de fouille de données dans une librairie de fouille de donnée disponible sur le Web à l'adresse [http ://www.philippe-fournier-viger.com/spmf/](http://www.philippe-fournier-viger.com/spmf/). La librairie a été téléchargé plus d'une centaine de fois par différents chercheurs et étudiants. Les implémentations des algorithmes de recherche de règles séquentielles présentés au chapitre 5 seront ajoutés à la librairie prochainement.

Limites et perspectives

Le travail comporte encore plusieurs éléments pouvant être améliorés. Cette section décrit les principaux. Des travaux sur ses aspects sont envisagés comme suite au doctorat.

Quantifier l'apprentissage. L'analyse cognitive fine a permis de percevoir l'impact positif des services tutoriels offerts aux apprenants grâce au modèle hybride dans CanadarmTutor. Comme suite au doctorat, il sera complémentaire d'effectuer une expérimentation empirique plus vaste pour quantifier l'impact sur l'apprentissage,

en utilisant un plus grand échantillon d'utilisateurs et des méthodes statistiques.

Extraction de généralités inter-exercices. L'approche d'acquisition automatique est conçue pour être appliquée pour chaque problème. Cela n'est pas une grande limitation puisque des ensembles d'exercices peuvent être prédéfinis pour bien des domaines. Toutefois, il serait intéressant d'étendre l'approche pour extraire des connaissances communes à plusieurs exercices.

Améliorer le modèle cognitif. Plusieurs aspects de la cognition spatiale n'ont pas été traités dans le modèle cognitif. En tenir compte permettrait une plus grande personnalisation. Un premier aspect sont les compétences spatiales et préférences spatiales de chaque individu. Les compétences spatiales sont des habiletés cognitives que chaque individu possède à différents degrés comme effectuer des rotations mentales d'objets ou imaginer des changements de perspectives [85, 98]. Hormis cela, d'autres aspects sont par exemple, la perception des mouvements et l'évaluation des distances.

Améliorer les algorithmes de fouille de données. Plusieurs améliorations pourraient être apportées à ces algorithmes telles que (1) en développer des versions incrémentales, (2) permettre d'extraire une représentation compacte et sans perte d'information des règles séquentielles, (3) utiliser des contraintes temporelles plus souples sur les trous dans les séquences (eg. [53]), (4) gérer des actions ayant plusieurs valeurs, et (5) optimiser l'implémentation des algorithmes. Aussi, une évaluation plus détaillée des performances des algorithmes pourraient être effectuée pour mieux comprendre leur performance.

Améliorer l'application des algorithmes de fouille de données. La façon dont les algorithmes de fouille de données sont appliqués dans CanadarmTutor peut aussi être amélioré. En particulier, il serait intéressant d'étudier l'effet de différentes divisions des problèmes en sous-problèmes (cf. 4.6) sur le nombre de motifs trouvés et leur qualité pour les recommandations aux utilisateurs.

Tester les solutions développées dans d'autres domaines. Une limite de la recherche est que ses produits ont été appliqués seulement dans CanadarmTutor.

Le modèle cognitif, le modèle hybride et l'approche d'acquisition pourraient être appliqués à d'autres domaines de connaissances pour les STI. De plus, les algorithmes de fouille de données pourraient être appliqués à d'autres tâches de fouille de données. Tout ceci permettrait d'en découvrir d'autres limites et utilités.

Améliorer les services tutoriels. Les services tutoriels présentés dans cette thèse et intégrés à CanadarmTutor sont présentés comme exemples de services tutoriels pouvant être offerts grâce au modèle hybride. Il est possible de les améliorer autant au niveau de leur présentation que de leur fonctionnement. Par exemple, des variantes de l'algorithme de reconnaissance de plan *RecognizePlan* présenté au chapitre 4 pourraient être testées pour voir si elles donnent de meilleurs résultats. Aussi, un aspect inexploré dans cette thèse est l'utilisation de stratégies pédagogiques complexes pour intégrer les différents services tutoriels.

Bibliographie

- (1) Abdessemed, A. et M. Najjar. 2007. « Toward a generic cognitive model of knowledge representation – a case study of problem solving in electrical circuits ». Dans *Young Researcher Track Proceedings of the 13th International Conference on Artificial Intelligence in Education*, p. 679–680.
- (2) Agrawal, R., T. Imielminsky, et A. Swami. 1993. « Mining association rules between sets of items in large databases ». Dans *Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data*, p. 207–216. ACM Press.
- (3) Agrawal, R. et J. Shafer. 1996. « Parallel mining of association rules. », *IEEE Transaction on Knowledge and Data Engineering*, vol. 8, no. 6, p. 962–969.
- (4) Agrawal, R. et R. Srikant. 1995. « Mining sequential patterns ». Dans *Proceedings of the 11th International Conference on Data Engineering*, p. 3–14. IEEE Computer Society.
- (5) Aleven, V. 2003. « Using background knowledge in case-based legal reasoning : a computational model and an intelligent learning environment », *Artificial Intelligence*, vol. 150, p. 183–237.
- (6) Aleven, V., K. Ashley, C. Lynch, et N. Pinkwart., éditeurs. 2006a. *Proceedings of the Intelligent Tutoring Systems for Ill-Defined Domains Workshop at ITS 2006*.
- (7) Aleven, V., K. Ashley, C. Lynch, et N. Pinkwart, éditeurs. 2007. *Proceedings of the Workshop on AIED applications in ill-defined domains at AIED 2007*.
- (8) Aleven, V., K. Ashley, C. Lynch, et N. Pinkwart, éditeurs. 2008. *Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains : Assessment and Feedback in Ill-Defined Domains at ITS 2008*.

- (9) Aleven, V., K. Ashley, C. Lynch, et N. Pinkwart, éditeurs. 2010. *Proceedings of the Workshop on ITS 2010 Workshop on ill-defined domains*.
- (10) Aleven, V., B. M. McLaren, J. Sewall, et K. Koedinger. 2006b. « The cognitive tutor authoring tools (CTAT) : Preliminary evaluation of efficiency gains ». Dans *Proceedings of 8th International Conference on Intelligent Tutoring Systems*, p. 61–70. Springer.
- (11) Aleven, V., B. M. McLaren, J. Sewall, et K. R. Koedinger. 2009. « A new paradigm for intelligent tutoring systems : Example-tracing tutors », *International Journal of Artificial Intelligence in Education*, vol. 19, no. 2, p. 105–154.
- (12) Anderson, J. R. 1993. *Rules of the mind*. Hillsdale : Lawrence Erlbaum Associates.
- (13) Anderson, J. R., A. T. Corbett, K. R. Koedinger, et R. Pelletier. 1995. « Cognitive tutors : Lessons learned », *Journal of Learning Science*, vol. 4, no. 2, p. 167–207.
- (14) Anderson, J. R. et S. Douglass. 2001. « Tower of hanoi : Evidence for the cost of goal retrieval », *Journal of Experimental Psychology : Learning, Memory and Cognition*, vol. 27, no. 6, p. 1331–1346.
- (15) Antunes, C. 2008. « Acquiring background knowledge for intelligent tutoring systems ». Dans *Proceedings of the 2nd International Conference on Educational Data Mining*, p. 18–27.
- (16) Ashley, K. D., R. Desai, et J. M. Levine. 2002. « Teaching case-based argumentation concepts using dialectic arguments vs. didactic explanations ». Dans *Proceedings of the 6th International Conference on Intelligent Tutoring Systems*, p. 585–595. Springer.
- (17) A. Terrazas, M. Krause and, P. Lipa, K. M. Gothard, C. A. Barnes, et B. L. McNaughton. 2005. « Self-Motion and the Hippocampal Spatial Metric », *Journal of Neuroscience*, vol. 25, no. 35, p. 8085–8096.

- (18) Barnes, T. et J. Stamper. 2008. « Toward automatic hint generation for logic proof tutoring using historical student data ». Dans *Proceedings of the 9th International Conference on Intelligent Tutoring Systems*, p. 373–382. Springer.
- (19) Belghith, K., F. Kabanza, L. Hartman, et R. Nkambou. 2006. « Anytime dynamic path-planning with flexible probabilistic roadmaps ». Dans *Proceedings of the 2006 IEEE International Conference on Robotics and Automation*, p. 2372–2377. IEEE.
- (20) Blessing, S. B. 1997. « A programming by demonstration authoring tool for model-tracing tutors », *International Journal of Artificial Intelligence in Education*, vol. 8, p. 233–261.
- (21) Bohbot, V. D., G. Laria, et M. Petrides. 2004. « Hippocampal function and spatial memory : Evidence from functional neuroimaging in healthy participants and performance of patients with medial temporal lobe resections », *Neuropsychology*, vol. 18, no. 3, p. 418–425.
- (22) Boonjing, V. et P. Songram. 2007. « Efficient algorithms for mining closed multidimensional sequential patterns ». Dans *Proceedings of the Fourth International Conference on Fuzzy Systems and Knowledge Discovery*. T. 2, p. 749–753. IEEE Computer Society.
- (23) Bouchard, F. 2005. « Guides de conception d'interfaces pour les systèmes tutoriels intelligents ». Mémoire, Université de Sherbrooke.
- (24) Burgess, N. 2006. « Spatial memory : how egocentric and allocentric combine », *Trends Cognitive Sciences*, vol. 10, no. 12, p. 551–557.
- (25) Burton, R. R. 1988. *The Environment Module of Intelligent Tutoring Systems*. Dans Polson, M. C. et J. J. Richardson, éditeurs, Coll. « *Foundations of Intelligent Tutoring Systems* », p. 109–142. Hillsdale, NJ : Erlbaum.
- (26) Byrne, R. M. J. et P. N. Johnson-Laird. 1989. « Spatial reasoning », *Journal of Memory and Language*, vol. 28, p. 564–575.

- (27) Carr, B. et I. P. Goldstein. 1977. Overlays : a theory of modeling for computer-aided instruction. AI Memo no. 406, MIT, Cambridge, Massachusetts..
- (28) Carruth, D., B. Robbins, M. Thomas, A. Morais, M. Letherwood, et K. Nebel. 2006. « Symbolic model of perception in dynamic 3D environments ». Dans *Proceedings of the 25th Army Science Conference*, p. 1-8.
- (29) Chen, G., X. Wu, et X. Zhu. 2005. « Sequential pattern mining in multiple streams ». Dans *Proceedings of the 5th International Conference on Data Mining*, p. 585-588. IEEE Computer Society.
- (30) Cheung, D. W., J. Han, V. Ng, et Y. Wong. 1996. « Maintenance of discovered association rules in large databases : An incremental updating technique ». Dans *Proceedings of the 12th International Conference on Data Engineering*, p. 962-969. IEEE Computer Society.
- (31) Chou, C.-Y., T.-W. Chan, et C.-J. Lin. 2003. « Redefining the learning companion : The past, present, and future of educational agents », *Computers and Education*, vol. 40, p. 255-269.
- (32) Clancey, W. 1984. *Use of MYCIN's rules for tutoring*. Dans Buchanan, B. et E. H. Shortliffe, éditeurs, Coll. « *Rule-Based Expert Systems. Reading* », p. 464-489. Addison-Wesley.
- (33) Corbett, A. T. et J. R. Anderson. 1992. *LISP Intelligent Tutoring System : Research in Skill Acquisition*. Dans Larkin, J. H. et R. W. Chabay, éditeurs, Coll. « *Computer-Assisted Instruction and Intelligent Tutoring Systems : Shared Goals and Complementary Approaches* », chapitre 3, p. 73-110. Lawrence Earlbaum.
- (34) Cournoyer, D. 2003a. Rapport général sur la semaine de formation « MSS MRO Part 1 » à l'agence spatiale canadienne. Rapport interne. Laboratoire Planariat. Université de Sherbrooke.
- (35) Cournoyer, D. 2003b. Technical note for the training week MSS MRO Part 1

- at the canadian space agency. Rapport interne. Laboratoire Planart. Université de Sherbrooke.
- (36) Das, G., K.-I. Lin, H. Mannila, G. Renganathan, et P. Smyth. 1998. « Rule discovery from time series ». Dans *Proceedings of the 4th International Conference on Knowledge Discovery and Data Mining*, p. 16–22. AAAI Press.
 - (37) Deogun, J. et L. Jiang. 2005. « Prediction mining – an approach to mining association rules for prediction ». Dans *Proceedings of the tenth International Conference on Rough Sets, Fuzzy Sets, Data Mining, and Granular Computing*, p. 98–108. Springer.
 - (38) Diez, M., D. A. Boehm-Davis, et R. W. Holt. 2002. « Model-based predictions of interrupted checklists ». Dans *Proceedings of the Human Factors and Ergonomics Society 46th Annual Meeting*, p. 250–254.
 - (39) Dragon, T., B. P. Woolf, D. Marshall, et T. Murray. 2006. « Coaching within a domain independent inquiry environment. ». Dans *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, p. 144–153. Springer.
 - (40) Dubois, D. 2006. Notes et observations sur la formation des astronautes à la manipulation du canadarm2 dans les locaux de l'agence spatiale canadienne (13 au 15 mars 2006). Rapport interne. Laboratoire GDAC. Université du Québec à Montréal.
 - (41) Dubois, D., P. Poirier, et R. Nkambou. 2007. « What does consciousness bring to CTS? ». Dans *Proceedings of the 2007 AAAI Fall Symposium*, p. 803–806. AAAI Press.
 - (42) Dugdale, S. 1992. *The design of computer-based mathematics education*. Dans Larkin, J. H. et R. W. Chabay, éditeurs, Coll. « *Computer-Assisted Instruction and Intelligent Tutoring Systems : Shared Goals and Complementary Approaches* », chapitre 1, p. 11–45. Lawrence Earlbaum.
 - (43) Easterday, M. W. 2010. *Policy World : A cognitive game for teaching deliberation*. Dans Pinkwart, N. et B. McLaren, éditeurs, Coll. « *Educational technologies for teaching argumentation skills* », p. sous presse. Bentham Science Publishers.

- (44) Easterday, M. W., V. Aleven, et R. Scheines. 2007. « The logic of babel : Causal reasoning from conflicting sources ». Dans *Proceedings of the Workshop on AIED applications in ill-defined domains at AIED 2007*.
- (45) Ekstrom, A. D., M. J. Kahana, J. B. Caplan, T. A. Fields, E. A. Isham, E. L. Newman, et I. Fried. 2003. « Cellular networks underlying human spatial navigation », *Nature*, vol. 424, p. 184–187.
- (46) Etienne, A. S., R. Maurer, et V. Séguinot. 1996. « Path integration in mammals and its interaction with visual landmarks », *Journal of experimental Biology*, vol. 199, p. 201–209.
- (47) Faghihi, U., P. Fournier-Viger, R. Nkambou, et P. Poirier. 2009a. « A generic episodic learning model implemented in a cognitive agent by means of temporal pattern mining ». Dans *Proceedings of the 22nd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, p. 545–555. Springer.
- (48) Faghihi, U., P. Fournier-Viger, R. Nkambou, et P. Poirier. 2010. « The combination of a causal learning and an emotional learning mechanism for an improved cognitive tutoring agent ». Dans *Proceedings of the 23rd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, p. 438–449. Springer.
- (49) Faghihi, U., P. Fournier-Viger, R. Nkambou, P. Poirier, et A. Mayers. 2009b. « How emotional mechanism helps episodic learning in a cognitive agent ». Dans *Proceedings of the 2009 IEEE Symposium on Intelligent Agents*, p. 23–30.
- (50) Faghihi, U., P. Poirier, D. Dubois, et R. Nkambou. 2008. « How emotional mechanism learn and helps other types of learning in a cognitive agent ». Dans *Proceedings of the 2008 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology*. T. 3, p. 606–611. IEEE.
- (51) Farquhar, A., R. Fikes, et J. Rice. 1997. « The ontolingua server : a tool for col-

- laborative ontology construction. », *International Journal Human-Computer Studies*, vol. 46, no. 6, p. 707–727.
- (52) Feigenbaum, J. D. et R. G. Moris. 2004. « Allocentric versus egocentric spatial memory after unilateral temporal lobectomy in humans », *Neuropsychology*, vol. 18, no. 3, p. 462–472.
- (53) Fiot, C., A. Laurent, et M. Teisseire. 2009. « Softening the blow of frequent sequence analysis : soft constraints and temporal accuracy. », *International Journal of Web Engineering Technologies*, vol. 5, no. 1, p. 24–47.
- (54) Fournier-Viger, P. 2005. « Un modèle de représentation des connaissances à trois niveaux de sémantique pour les systèmes tutoriels intelligents ». Mémoire de maîtrise, Université de Sherbrooke. 194 pages.
- (55) Fournier-Viger, P., U. Faghihi, R. Nkambou, et E. M. Nguifo. 2010a. « Exploiting sequential patterns found in users' solutions and virtual tutor behavior to improve assistance in its », *Educational Technology & Society*, vol. 13, no. 1, p. 13–24.
- (56) Fournier-Viger, P., M. Najjar, et A. Mayers. 2005. « Combining the learning objects paradigm with cognitive modelling theories - a novel approach for knowledge engineering ». Dans *Proceedings of the ITI 3rd International Conference on Information & Communication Technology*, p. 565–578.
- (57) Fournier-Viger, P., M. Najjar, A. Mayers, et R. N. Mayers. 2006a. « A cognitive and logic based model for building glass-box learning objects », *Interdisciplinary Journal of Knowledge and Learning Objects*, vol. 2, p. 77–94.
- (58) Fournier-Viger, P., M. Najjar, A. Mayers, et R. Nkambou. 2006b. « From black-box learning objects to glass-box learning objects. ». Dans Ikeda, M., K. D. Ashley, et T.-W. Chan, éditeurs, *Intelligent Tutoring Systems*. T. 4053, série *Lecture Notes in Computer Science*, p. 258–267. Springer.

- (59) Fournier-Viger, P., R. Nkambou, U. Faghihi, et E. M. Nguifo. 2009a. *Mining Temporal Patterns to Improve Agents Behavior : Two Case Studies*. Dans Cao, L., éditeur, Coll. « *Data Mining and Multiagent Integration* », p. 77–92. Springer.
- (60) Fournier-Viger, P., R. Nkambou, et A. Mayers. 2008a. « Evaluating spatial knowledge through problem-solving in virtual learning environments ». Dans *Proceedings Third European Conference on Technology Enhanced Learning*, p. 15–26. Springer.
- (61) Fournier-Viger, P., R. Nkambou, et A. Mayers. 2008b. « Evaluating spatial representations and skills in a simulator-based tutoring system », *IEEE Transactions on Learning Technologies*, vol. 1, no. 1, p. 63–74.
- (62) Fournier-Viger, P., R. Nkambou, et A. Mayers. 2008c. « A framework for evaluating semantic knowledge in problem-solving-based intelligent tutoring systems. ». Dans Wilson, D. et H. C. Lane, éditeurs, *Proc. 23th FLAIRS Conference*, p. 409–414. AAAI Press.
- (63) Fournier-Viger, P., R. Nkambou, A. Mayers, et D. Dubois. 2007. « Automatic evaluation of spatial representations for complex robotic arms manipulations. ». Dans Spector, J. M., D. G. Sampson, T. Okamoto, Kinshuk, S. A. Cerri, M. Ueno, et A. Kashiara, éditeurs, *ICALT*, p. 279–281. IEEE Computer Society.
- (64) Fournier-Viger, P., R. Nkambou, et E. M. Nguifo. 2008a. « A data mining framework for the acquisition of procedural knowledge in intelligent tutoring systems ». Dans *Proceedings of the 16th International Conference on Computers in Education*, p. 153–157.
- (65) Fournier-Viger, P., R. Nkambou, et E. M. Nguifo. 2008b. « A knowledge discovery framework for learning task models from user interactions in intelligent tutoring systems ». Dans *Proceedings of the 7th Mexican International Conference on Artificial Intelligence*, p. 765–778. Springer.
- (66) Fournier-Viger, P., R. Nkambou, et E. M. Nguifo. 2008c. « A sequential pattern mining algorithm for extracting partial problem spaces from logged user interactions ».

- Dans *Proceedings of the 3rd International Workshop on Intelligent Tutoring Systems in Ill-Defined Domain, at ITS 2008*, p. 46–55.
- (67) Fournier-Viger, P., R. Nkambou, et E. M. Nguifo. 2009. « Exploiting partial problem spaces learned from users' interactions to provide key tutoring services in procedural and ill-defined domains ». Dans *Proceedings of the 14th International Conference on Artificial Intelligence and Education*, p. 383–390. IOS Press.
 - (68) Fournier-Viger, P., R. Nkambou, et E. M. Nguifo. 2010a. *Building Intelligent Tutoring Systems for Ill-Defined Domains*. Dans Nkambou, R., R. Mizoguchi, et J. Bourdeau, éditeurs, Coll. « *Advances in Intelligent Tutoring Systems* », p. 81–101. Springer.
 - (69) Fournier-Viger, P., R. Nkambou, et E. M. Nguifo. 2010b. « Cmrules : An efficient algorithm for mining sequential rules common to several sequences ». Dans *Proceedings of the 23th International Florida Artificial Intelligence Research Society Conference*, p. 410–415. AAAI Press.
 - (70) Fournier-Viger, P., R. Nkambou, et E. M. Nguifo. 2010c. *Learning Procedural Knowledge from User Solutions To Ill-Defined Tasks in a Simulated Robotic Manipulator*. Dans Romero, C., éditeur, Coll. « *Handbook of Educational Data Mining* », p. 451–465. CRC Press.
 - (71) Fournier-Viger, P., R. Nkambou, E. M. Nguifo, et U. Faghihi. 2009b. « Building agents that learn by observing other agents performing a task - a sequential pattern mining approach ». Dans *Proceedings of the 22nd International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems*, p. 279–284. Springer.
 - (72) Fournier-Viger, P., R. Nkambou, E. M. Nguifo, et A. Mayers. 2010b. « Its in ill-defined domains, toward hybrid approaches ». Dans *Proceedings of the 10th International Conference on Intelligent Tutoring Systems*, p. 749–751. Springer.
 - (73) Frye, D. et E. Soloway. 1987. « Interface design : a neglected issue in educational

- software ». Dans *CHI '87 : Proceedings of the SIGCHI/GI conference on Human factors in computing systems and graphics interface*, p. 93–97. ACM Press.
- (74) Garofalakis, M. N., R. Rastogi, et K. Shim. 1999. « SPIRIT : Sequential pattern mining with regular expression constraints ». Dans *Proceedings of 25th International Conference on Very Large Data Bases*, p. 223–234. Morgan Kaufmann.
- (75) Gouda, K., M. Hassaan, et M. J. Zaki. 2010. « Prism : An effective approach for frequent sequence mining via prime-block encoding », *Journal of Computers and Systems Sciences*, vol. 76, no. 1, p. 88–102.
- (76) Graesser, A., P. Wiemer-Hastings, K. Wiemer-Hastings, D. Harter, et N. Person. 2000. « Using latent semantic analysis to evaluate the contributions of students in autotutor », *Interactive Learning Environments*, vol. 8, p. 149–169.
- (77) Gunzelmann, G. et R. D. Lyon. 2006. « Mechanisms of human spatial competence ». Dans *Proceedings of the Spatial Cognition V Conference*, p. 288–307. Springer.
- (78) Halford, G. S., R. Baker, J. E. McCredden, et J. D. Bain. 2005. « How many variables can humans process ? », *Psychological Science*, vol. 16, no. 1, p. 70–76.
- (79) Hamilton, H. J. et K. Karimi. 2005. « The TIMERS II algorithm for the discovery of causality ». Dans *Proceedings of the 9th Pacific-Asia Conference on Knowledge Discovery and Data Mining*, p. 744–750.
- (80) Han, J. et M. Kamber. 2006. *Data Mining : concepts and techniques*. Morgan Kaufmann Publisher.
- (81) Han, J., J. Pei, B. Mortazavi-Asl, Q. Chen, U. Dayal, et M. Hsu. 2000. « Freespan : frequent pattern-projected sequential pattern mining. ». Dans *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, p. 355–359.
- (82) Han, J., J. Pei, Y. Yin, et R. Mao. 2004. « Mining frequent patterns without can-

- didate generation : A frequent-pattern tree approach », *Data Mining and Knowledge Discovery*, vol. 8, no. 1, p. 53–87.
- (83) Harms, S. K., J. Deogun, et T. Tadesse. 2002. « Discovering sequential association rules with constraints and time lags in multiple sequences. ». Dans *Proceedings of the 13th International Symposium on Methodologies for Intelligent Systems*, p. 373–376.
- (84) Harrison, M. et C. D. Schunn. 2003. « ACT-R/S : Look ma, no cognitive map ! ». Dans *Proceedings of the Fifth International Conference on Cognitive Modeling*, p. 129–134. Universitäts-Verlag Bamger.
- (85) Hegarty, M. et D. Waller. 2004. « A dissociation between mental rotation and perspective-taking spatial abilities », *Intelligence*, vol. 32, p. 175–191.
- (86) Hegland, M. 2007. « The apriori algorithm – a tutorial », *Mathematics and Computation in Imaging Science and Information Processing*, vol. 11, p. 209–262.
- (87) Hirate, Y. et H. Yamana. 2006. « Generalized sequential pattern mining with item intervals », *Journal of Computers*, vol. 1, no. 3, p. 51–60.
- (88) Hodhod, R. et D. Kudenko. 2008. « Interactive narrative and intelligent tutoring for ethics domain ». Dans *Proceedings of the ITS for Ill-Defined Domains Workshop at ITS 2008*, p. 13–21.
- (89) Hotte, R. et P. Leroux. 2003. « Technologies et formation à distance », *revue STICEF*, vol. 10. Numéro spécial : Technologies et Formation à distance.
- (90) Hsieh, Y. L., D.-L. Yang, et J. Wu. 2006. « Using data mining to study upstream and downstream causal relationship in stock market ». Dans *Proceedings of the 2006 Joint Conference on Information Sciences*.
- (91) Jarvis, M., G. Nuzzo-Jones, et N. T. Heffernan. 2005. « Applying machine learning techniques to rule generation in intelligent tutoring systems ». Dans *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, p. 541–553. Springer.

- (92) Jonassen, I., J. F. Collins, et D. G. Higgins. 1995. « Finding flexible patterns in unaligned protein sequences », *Protein Science*, vol. 4, no. 8, p. 1587–1595.
- (93) Kabanza, F., R. Nkambou, et K. Belghith. 2005. « Path-planning for autonomous training on robot manipulators in space ». Dans *Proceedings of the 19th International Joint Conference on Artificial Intelligence*. Professional Book Center.
- (94) Kay, J. 2000. « Stereotypes, student models and scrutability ». Dans *Proceedings of the 5th International Conference on Intelligent Tutoring Systems*, p. 19–30.
- (95) Kodaganallur, V., R. Weitz, et D. Rosenthal. 2006. « An assessment of constraint-based tutors : A response to mitrovic and ohlsson's critique of "a comparison of model-tracing and constraint-based intelligent tutoring paradigms" », *International Journal of Artificial Intelligence in Education*, vol. 16, p. 291–321.
- (96) Koedinger, K., V. Aleven, N. Heffernan, B. M. McLaren, et M. Hockenberry. 2004. « Opening the door to non-programmers : Authoring intelligent tutor behavior by demonstration ». Dans *Proceedings of the 7th International Conference on Intelligent Tutoring Systems*, p. 162–174. Springer.
- (97) Koedinger, K. R., J. R. Anderson, W. H. Hadley, et M. A. Mark. 1997. « Intelligent tutoring goes to school in the big city », *International Journal of Artificial Intelligence in Education*, vol. 1, no. 8, p. 30–43.
- (98) Kozhevnikov, M. et M. Hegarty. 2001. « A dissociation between object manipulation spatial ability and spatial orientation ability », *Memory & Cognition*, vol. 29, p. 754–755.
- (99) Kriatofic, A. et M. Bielikova. 2005. « Improving adaptation in web-based educational hypermedia by means of knowledge discovery ». Dans *Proceedings of the 16th ACM Conference on Hypertext and Hypermedia*, p. 184–192. ACM Press.
- (100) Langley, P. et S. Ohlsson. 1984. « Automated cognitive modeling ». Dans *Proceedings of the 4th National Conference on Artificial Intelligence*, p. 193–197. AAAI Press.

- (101) Lin, M.-Y. et S.-C. Hsueh. 2008. « Mining closed sequential patterns with time constraints », *Journal of Information Science and Engineering*, vol. 24, no. 1, p. 33–46.
- (102) Lin, M.-Y., S.-C. Hsueh, et C.-W. Chang. 2006. « Mining closed sequential patterns with time constraints ». Dans *Proceedings of Workshop on Software Engineering, Databases, and Knowledge Discovery at International Computer Symposium 2006*.
- (103) Lo, D., S.-C. Khoo, et J. Lindsay. 2008. « Mining and ranking generators of sequential patterns ». Dans *Proceedings of the 8th SIAM International Conference on Data Mining*, p. 553–564.
- (104) Lo, D., S.-C. Khoo, et L. Wong. 2009. « Non-redundant sequential rules - theory and algorithms », *Information Systems*, vol. 34, no. 4, p. 438–453.
- (105) Lynch, C., K. Ashley, V. Aleven, et N. Pinkwart. 2006. « Defining ill-defined domains ». Dans *Proceedings of the Workshop on Intelligent Tutoring Systems for Ill-Defined Domains at ITS 2006*, p. 1–10.
- (106) MacQueen, J. 1967. « Some methods for classification and analysis of multivariate observations ». Dans *Proceedings of the 5th Berkeley Symposium on Mathematic Statistics and Probability*, p. 281–297.
- (107) Mannila, H., H. Toivonen, et A. I. Verkana. 1997. « Discovery of frequent episodes in event sequences », *Data Mining and Knowledge Discovery*, vol. 1, no. 1, p. 259–289.
- (108) Matsuda, N., W. W. Cohen, J. Sewall, G. Lacerda, et K. R. Koedinger. 2007. « Predicting students' performance with simstudent : Learning cognitive skills from observation ». Dans *Proceedings of the 13th International Conference on Artificial Intelligence in Education*, p. 467–478. IOS Press.
- (109) Mayers, A., B. Lefebvre, et C. Frasson. 2001. « Miace, a human cognitive architecture », *SIGCUE Outlook*, vol. 27, no. 2, p. 61–77.
- (110) McLaren, B. M., K. R. Koedinger, M. Schneider, A. Harrer, et L. Bollen. 2004. « Bootstrapping novice data : Semi-automated tutor authoring using student log

- files ». Dans *Proceedings of the Workshop on Analyzing Student-Tutor Interaction Logs to Improve Educational Outcomes at ITS 2004*.
- (111) McNamara, T. 2003. « How are the locations of objects in the environment represented in memory ? ». Dans *Proceedings of the Spatial Cognition III Conference*, p. 174–191. Springer.
- (112) Mcnaughton, B. L., F. P. Battaglia, O. Jensen, E. I. Moser, et M.-B. Moser. 2006. « Path integration and the neural basis of the 'cognitive map' », *Nature Reviews Neuroscience*, vol. 7, no. 8, p. 663–678.
- (113) Milik, N., A. Mitrovic, et M. Grimley. 2007. « Fitting spatial ability into its development ». Dans *Proceedings of the 13th International Conference on Artificial Intelligence in Education*, p. 617–619. IOS Press.
- (114) Mitrovic, A., M. Mayo, P. Suraweera, et B. Martin. 2001. « Constraint-based tutors : a success story ». Dans *Proceedings of the 14th International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems*, p. 931–940. Springer.
- (115) Mitrovic, A. et A. Weerasinghe. 2009. « Revisiting ill-definedness and the consequences for itss ». Dans *Proceedings of the 14th International Conference on Artificial Intelligence in Education*, p. 375–382. IOS Press.
- (116) Mittelstaedt, M. et H. Mittelstaedt. 1980. « Homing by path integration in a mammal », *Naturwissenschaften*, vol. 67, p. 566–567.
- (117) Morales, R. et A. S. Aguera. 2002. « Dynamic sequencing of learning objects ». Dans *Proceedings of 2nd International Conference on Advanced Learning Technologies*, p. 502–506. IEEE Computer Society.
- (118) Morganti, F., A. Gaggioli, L. Strambi, M. L. Rusconi, et G. Riva. 2007. « A virtual reality extended neuropsychological assessment for topographical disorientation : a feasibility study », *Journal of NeuroEngineering and Rehabilitation*, vol. 4, no. 27.

- (119) Moritz, S. et G. Blank. 2008. « Generating and evaluating object-oriented design for instructors and novice students ». Dans *Proceedings of the ITS for Ill-Defined Domains Workshop at ITS2008*, no 35-45.
- (120) Nadel, L. et O. Hardt. 2004. « The spatial brain », *Neuropsychology*, vol. 18, no. 3, p. 473–476.
- (121) Nanayakkara, T., A. Dissanayake, P. Mahipala, et G. Sanjaya. 2008. *A Human-Animal-Robot Cooperative System for Anti-Personal Mine Detection*. Dans Habib, M., éditeur, Coll. « *Humanitarian Demining : Innovative Solutions and the Challenges of Technology* », p. 347–366. ARS publishers.
- (122) Newell, A. 1980. *Reasoning, problem solving and decision processes : The problem space as a fundamental category*. Dans Nickerson, R., éditeur, Coll. « *Attention and performance, VIII* ». Lawrence Erlbaum Associates.
- (123) Newell, A. et H. A. Simon. 1972. *Human Problem Solving*. Prentice-Hall, Inc.
- (124) Nkambou, R., K. Belghith, et F. Kabanza. 2006. « An approach to intelligent training on a robotic simulator using an innovative path-planner ». Dans *Proceedings of the 8th International Conference on Intelligent Tutoring Systems*, p. 645–654. Springer.
- (125) Nkambou, R., P. Fournier-Viger, et E. M. Nguifo. 2009. « Improving the behavior of intelligent tutoring agents with data mining », *IEEE Intelligent Systems*, vol. 24, no. 3, p. 46–53.
- (126) Nkambou, R., P. Fournier-Viger, et E. M. Nguifo. 2010. « Learning task models in ill-defined domain using an hybrid knowledge discovery framework », *Knowledge-Based Systems*, p. sous presse.
- (127) Nkambou, R., E. M. Nguifo, O. Couturier, et P. Fournier-Viger. 2007a. « A framework for problem-solving knowledge mining from users' actions ». Dans *Proceedings of the 13th International Conference on Artificial Intelligence in Education*, p. 623–625.

- (128) Nkambou, R., E. M. Nguifo, O. Couturier, et P. Fournier-Viger. 2007b. « Problem-solving knowledge mining from users' actions in an intelligent tutoring system ». Dans *Proceedings of the 19th Canadian Conference on Artificial Intelligence*, p. 394–404. Springer.
- (129) Nkambou, R., E. M. Nguifo, et P. Fournier-Viger. 2008. « Using knowledge discovery techniques to support tutoring in an ill-defined domain. ». Dans *Proceedings of the 9th International Conference on Intelligent Tutoring Systems*, p. 395–405. Springer.
- (130) Ogan, A., R. Wylie, et E. Walker. 2006. « The challenges in adapting traditional techniques for modeling student behavior in ill-defined domains ». Dans *Proceedings of the Intelligent Tutoring Systems for Ill-Defined Domains Workshop at ITS2006*, p. 29–37.
- (131) O'Keefe, J. et L. Nadel. 1978. *The hippocampus as a cognitive map*. Oxford :Clarendon.
- (132) Pasquier, N., Y. Bastide, R. Taouil, et L. Lakhal. 1999. « Closed set based discovery of small covers for association rules ». Dans *15èmes Journées Bases de Données Avancées*, p. 361–381.
- (133) Pazzaglia, F. et R. D. Beni. 2001. « Strategies of processing spatial information in survey and landmark-centred individuals », *European Journal of Cognitive Psychology*, vol. 13, no. 4, p. 493–508.
- (134) Pei, J., J. Han, S. Member, B. Mortazavi-asl, J. Wang, H. Pinto, Q. Chen, U. Dayal, et M. chun Hsu. 2004. « Mining sequential patterns by pattern-growth : The prefixspan approach », *IEEE Transactions on Knowledge and Data Engineering*, vol. 16, no. 10, p. 1–17.
- (135) Perera, D., J. Kay, I. Koprinska, K. Yacef, et O. Zaiane. 2008. « Clustering and sequential pattern mining of online collaborative learning data », *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 6, p. 759–772.

- (136) Pinto, H., J. Han, J. Pei, K. Wang, Q. Chen, et U. Dayal. 2001. « Multi-dimensional sequential pattern mining ». Dans *Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management*, p. 81–88. ACM.
- (137) Redish, D., A. N. Elga, et D. S. Touretzky. 1996. « A coupled attractor model of the rodent head direction system », *Network : Computation in Neural Systems*, vol. 7, no. 4, p. 671–685.
- (138) Riccuci, S., A. Carbonaro, et G. Casadei. 2007. « Knowledge acquisition in intelligent tutoring system : A data mining approach ». Dans *Proceedings of the 6th Mexican Conference on Artificial Intelligence*, p. 1195–1205.
- (139) Romero, C., S. Gutierrez, M. Freire, et S. Ventura. 2008. « Mining and visualizing visited trails in web-based educational systems ». Dans *Proceedings of the 2nd International Conference on Educational Data Mining*, p. 182–186.
- (140) Shelton, L. et J. Gabrieli. 2004. « Neural correlates of encoding space from route and survey perspectives », *Neuropsychology*, vol. 18, no. 3, p. 442–449.
- (141) Simon, H. A. 1978. *Information-processing theory of human problem solving*. Dans Estes, W. K., éditeur, Coll. « *Handbook of learning and cognitive processes : Vol. 5. Human information* ». Hillsdale, NJ : Erlbaum.
- (142) Soller, A., A. Martinez-Monez, P. Jermann, et M. Muehlenbrock. 2005. « From mirroring to guiding : A review of state of the art technology for supporting collaborative learning. », *International Journal of Artificial Intelligence in Education*, vol. 15, no. 4, p. 261–290.
- (143) Songram, P., V. Boonjing, et S. Intakosum. 2006. « Closed multidimensional sequential pattern mining ». Dans *Proceedings of the 3rd International Conference on Information Technology*, p. 512–517.
- (144) Srikant, R. et R. Agrawal. 1996. « Mining sequential patterns : Generalizations and

- performance improvements ». Dans *Proceedings of the 5th International Conference on Extending Database Technology*, p. 3–17.
- (145) Star, J. R. 2000. « On the relationship between knowing and doing in procedural learning ». Dans *Proceedings of the Fourth International Conference on Learning Sciences*, p. 80–86.
- (146) Su, J.-M., S.-S. Tseng, W. Wang, J.-F. Weng, J. T. D. Yang, et W.-N. Tsai. 2006. « Learning portfolio analysis and mining for scorm compliant environment », *Educational Technology & Society*, vol. 9, no. 1, p. 262–275.
- (147) Sun, L. et X. Zhang. 2004. « Efficient frequent pattern mining on web logs ». Dans *Proceedings of APWeb 2004*, p. 533–542. Springer.
- (148) Suraweera, P., A. Mitrovic, et B. Martin. 2005. « A knowledge acquisition system for constraint-based intelligent tutoring systems ». Dans *Proceedings of the 12th International Conference on Artificial Intelligence in Education*, p. 638–645. IOS Press.
- (149) Suraweera, P., A. Mitrovic, et B. Martin. 2007. « Constraint authoring system : An empirical evaluation ». Dans *Proceedings of the 13th International Conference on Artificial Intelligence in Education*, p. 451–458. IOS Press.
- (150) Taricani, E. M. et R. B. Clariana. 2006. « A technique for automatically scoring open-ended concept maps », *Educational Technology Research & Development*, vol. 54, no. 1, p. 61–78.
- (151) Taube, J., R. Muller, et J. Ranck. 1990a. « Head-direction cells recorded from the postsubiculum in freely moving rats. 1. description and quantitative analysis », *The Journal of Neuroscience*, vol. 10, no. 2, p. 420–435.
- (152) Taube, J., R. U. Muller, et J. B. Ranck. 1990b. « Head-direction cells recorded from the postsubiculum in freely moving rats. 2. effects of environmental manipulations », *The Journal of Neuroscience*, vol. 10, no. 2, p. 436–447.

- (153) Templeman, J. N. et L. E. Sibert. 2006. *Immersive Simulation of Coordinated Motion in Virtual Environments : An application to Training Small Unit Military Tacti Techniques and Procedures*. Dans Allen, G., éditeur, Coll. « *Applied spatial cognition : from research to cognitive technology* », p. 339–372. NJ :Lawrence Erlbaum.
- (154) Tollman, E. 1948. « Cognitive maps in rats and men », *Psychological Review*, vol. 5, no. 4, p. 189–208.
- (155) Tulving, E. 1983. *Element of Episodic Memory*. Oxford : Clarendon Press.
- (156) Tversky, B. 1993. « Cognitive maps, cognitive collages, and spatial mental models ». Dans *Proceedings of the 1993 European Conference on Spatial Information Theory*, p. 14–24. Springer.
- (157) van Lent, M. et J. E. Laird. 2001. « Learning procedural knowledge through observation ». Dans *Proceedings of the First International Conference on Knowledge Capture*, p. 179–186.
- (158) VanLehn, K. 1988. *Student Modeling*. Dans Polson, M. C. et J. J. Richardson, éditeurs, Coll. « *Foundations of Intelligent Tutoring Systems* », p. 55–78. NJ :Hillsdale : Erlbaum.
- (159) VanLehn, K., C. Lynch, K. Schulze, R. Shelby, L. Taylor, D. Treacy, A. Weinstein, et M. Wintersgill. 2005. « The andes physics tutoring system : Lessons learned », *International Journal of Artificial Intelligence and Education*, vol. 15, p. 147–204.
- (160) Walker, E., A. Ogan, V. Aleven, et C. Jones. 2008. « Two approaches for providing adaptive support for discussion in an ill-defined domain ». Dans *Proceedings of the ITS for Ill-Defined Domains Workshop at ITS2008*, p. 1–11.
- (161) Wang, J. et J. Han. 2004. « Bide : Efficient mining of frequent closed sequences ». Dans *Proceedings of the 20th International Conference on Data Engineering*, p. 79–90.
- (162) Wang, J., J. Han, et C. Li. 2007. « Frequent closed sequence mining without

- candidate maintenance », *IEEE Transactions on Knowledge and Data Engineering*, vol. 19, no. 8, p. 1042–1056.
- (163) Wenger, E. 1987. *Artificial intelligence and tutoring systems : computational and cognitive approaches to the communication of knowledge*. Morgan Kaufmann Publishers Inc.
- (164) Woolf, B. 2009. *Building Intelligent Interactive Tutors : Student Centered Strategies for revolutionizing e-learning*. Morgan Kaufmann.
- (165) Yan, X., J. Han, et R. Afshar. 2003. « Clospan : Mining closed sequential patterns in large databases ». Dans *Proceedings of the Third SIAM International Conference on Data Mining*, p. 166–177. SIAM.
- (166) Yang, Z., Y. Wang, et M. Kitsuregawa. 2007. « LAPIN : Effective sequential pattern mining algorithms by last position induction for dense databases ». Dans *Proceedings of the 12th International Conference on Database Systems for Advanced Applications*, p. 1020–1023.
- (167) Yuan, D., K. Lee, H. Cheng, G. Krishna, Z. Li, X. Ma, Y. Zhou, et J. Han. 2008. « CISpan : Comprehensive incremental mining algorithms of closed sequential patterns for multi-versional software mining ». Dans *Proceedings of the 8th SIAM International Conference on Data Mining*, p. 84–95.
- (168) Zaki, M. J. 2000. « Scalable algorithms for association mining », *IEEE Transactions on Knowledge and Data Engineering*, vol. 12, no. 3, p. 372–390.
- (169) Zaki, M. J. 2001. « Spade : An efficient algorithm for mining frequent sequences », *Machine Learning Journal*, vol. 42, no. 1/2, p. 31–60.